commodore újság

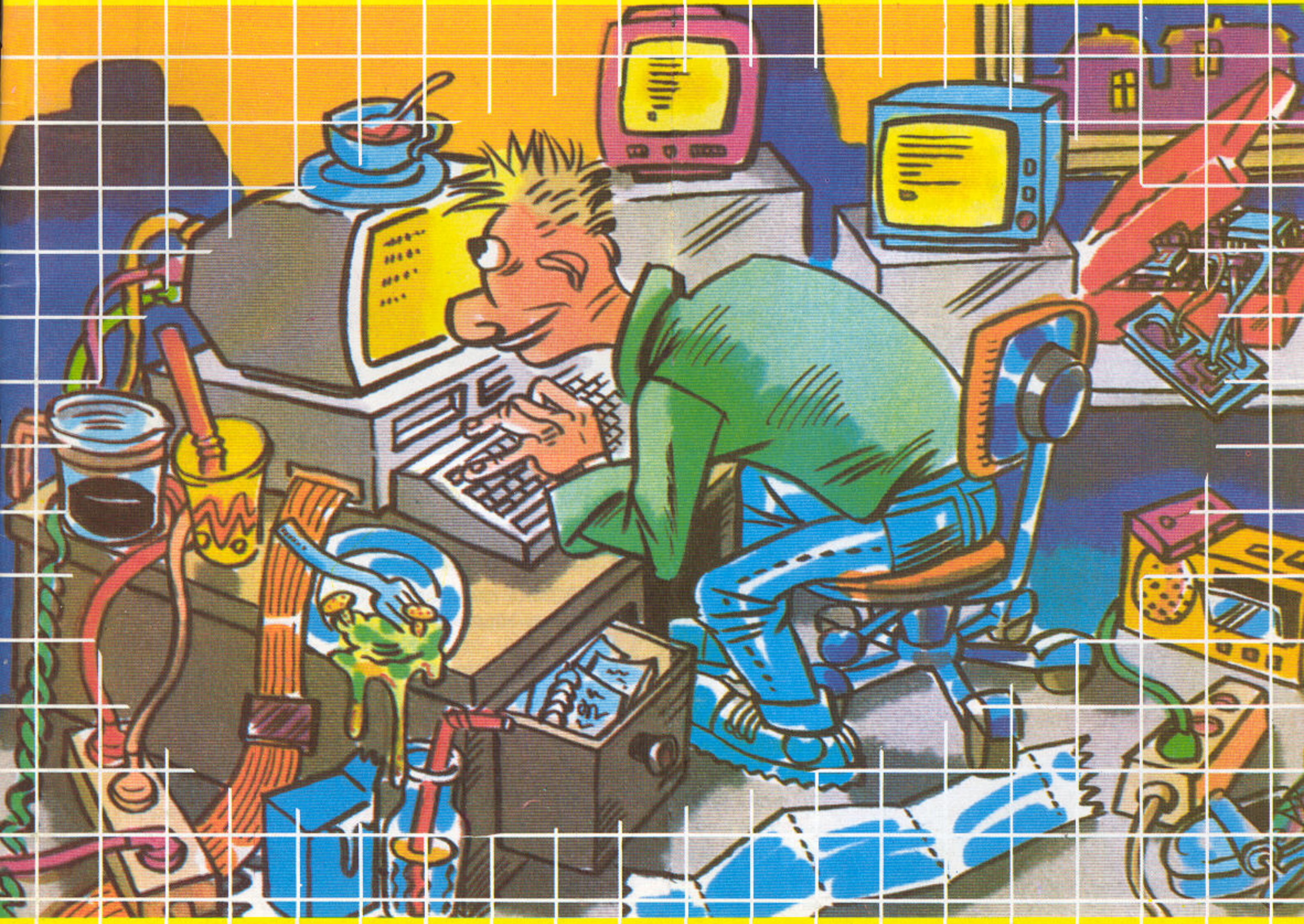
IX.évfolyam
1994/1

Az Országos Commodore Egyesület lapja

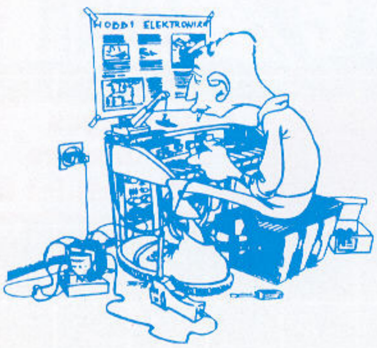
Othello
TCT program
Karakterező

Játék

Prince of Persia



BUÉK 1994



HOBBI ELEKTRONIKA

Szuper ajánlat!

Urbán István mérnök áramköreinek szaküzlete

Budapest VII., Dózsa György út 16. (Dózsa–Jobbágy sarok)
Nyitva: H–P 10–17-ig Tel./fax: 122-8892
(Zárás után üzenetrögzítő)

KÍNÁLATUNKBÓL

SZÁMÍTÁSTECHNIKA:

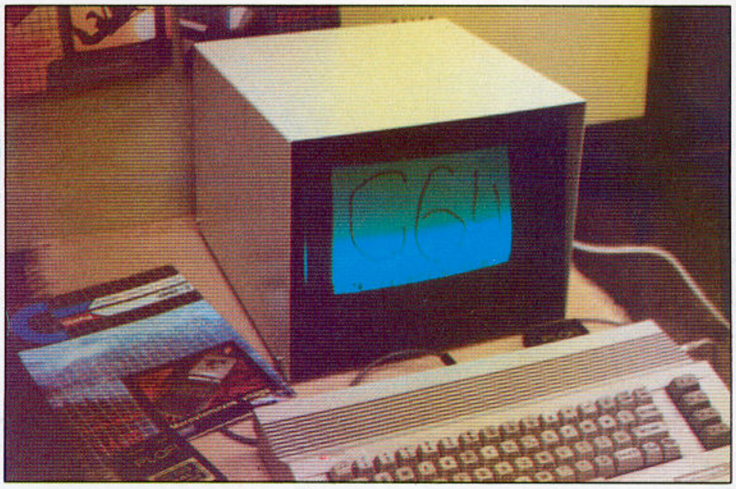
C64 BŐVÍTŐK	egys.	élesztve
PLOFI Datassette cartridge	1.200 Ft	1.500 Ft
PLOFI Help + cartridge	1.200 Ft	1.500 Ft
PLOFI FASTLOAD cartridge	1.200 Ft	1.500 Ft
PLOFI FASTLOAD + SPEEDTAPE	1.500 Ft	1.700 Ft
PLOFI SIMON'S cartridge	1.500 Ft	1.700 Ft
PLOFI JÁTÉK cartridge	1.500 Ft	1.700 Ft
ACTION REPLAY MK7 cartridge	—	3.450 Ft
ATOMIC POWER cartridge	—	3.450 Ft
FINAL III. cartridge	—	3.450 Ft
PAGEFOX DTP cartridge	3.200 Ft	3.900 Ft
MINI EPROMBANK cartridge	2.300 Ft	2.900 Ft
EPROMBANK 256 Kbájt	2.900 Ft	4.000 Ft
C64 DOKTOR teszt cartridge	2.300 Ft	3.000 Ft
EPROMÉGETŐ (2716-27512-ig)	3.400 Ft	4.500 Ft
IC TESZTER (kb. 180 TTL IC)	2.700 Ft	3.900 Ft
256K RAM BŐVÍTŐ RAM floppi	—	11.800 Ft
FÉNYCERUZA + szoftver	950 Ft	1.450 Ft
HANGDIGITALIZÁLÓ + szoftver	1.350 Ft	1.700 Ft
HANGKAPCSOLÓ + szoftver	700 Ft	990 Ft
DATASSETTE fejbeállító	390 Ft	—
CPU STOP + RESET	400 Ft	—
FREKVENCIAMÉRŐ 30 MHz	1.300 Ft	1.800 Ft
I/O BŐVÍTŐ 8255-tel	1.400 Ft	1.800 Ft
JOYSTICK automata	1.500 Ft	1.900 Ft
USER CENTRONICS kábel	750 Ft	—

IBM BŐVÍTŐK	egys.	élesztve
IBM IC TESZTER + szoftver	5.900 Ft	9.800 Ft
IBM EPROMÉGETŐ + szoftver	4.800 Ft	9.900 Ft
IBM EPROMÉGETŐ tápegység	1.200 Ft	↑ benne
IBM CODEC HANGKÁRTYA	3.800 Ft	4.800 Ft

IBM KOVOX	—	700 Ft
IBM AKTÍV KOVOX	1.600 Ft	—
IBM I/O KÁRTYA 48 vonal	3.500 Ft	—
IBM GAL PROGRAMOZÓ	6.500 Ft	8.500 Ft

TANGÓ PLD jogtiszta szoftver. *Kérjen árajánlatot!*
TANGÓ NYÁKTEVEZŐ jogtiszta szoftver
TTL, CMOS, TRANZISZTOR, DIÓDA, MEMÓRIA
katalógus floppylemezen, egységesen: 600 Ft/db

AKCIÓ! Amíg a készlet tart! **MONITORVÁSÁR**



Monochrom, zöld színű, 28 cm átlójú új monitorok C-64, Plus/4 stb. gépekhez.
Ára: tápegységgel csak 1700 Ft (ÁFA-val), tápegység nélkül 600 Ft (ÁFA-val)

Áraink a 25% ÁFA-t tartalmazzák!

Szaküzletemben vásárolhat a Commodore Újság HOBBI ELEKTRONIKA rovatában megjelent hardverleírásokhoz (működő minta alapján) panelt, egységcsomagot, vagy készre szerelt áramkört.

MŰSZERVÁSÁR
Bontott anyagok, használt műszerek böngészdéje



Állandóan változó készlet!

Vidéki olvasóknak segít a szerző levelező, egységcsomagküldő szolgáltatása: a megrendelt csomagot postán utánvéttel elküldöm. Telefonon/faxon és levélben is rendelhet.
A HOBBI ELEKTRONIKÁ-hoz nem kell hosszú levél. Rendelését néhány sorban, egyértelműen közölje.

Levélcím: 1656 Budapest Pf. 50.

Az üzletben megvásárolhatók az RT évkönyvei, a Rádiótechnika és a Hobby Elektronika korábbi számai.

SZERENCSEJÁTÉK RT

MIT, HOGYAN, HOL, MIKOR?

EGYESÜLETI ÜGYEK: Egyesületünknek tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az egyesület irodájában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

Pötyögőszolgálatunk valamint a szervizkedvezmény és az apróhirdetés lehetősége tagjaink rendelkezésére áll.

A DEÁKPÁHOLY tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy egy évre 890 Ft.

A PLUSZPÁHOLY tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 2050 Ft.

A SZUPERPÁHOLY tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15x3 db vásárlási utalványt is. Az éves tagsági díj 24 000 Ft.

ÜGYFÉLFOGADÁS: Minden kedden és csütörtökön 12—16 óra között várjuk tagjainkat és az érdeklődőket.

PÖTYÖGŐSZOLGÁLAT: Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az egyesület irodájában vagy postai utánvéttel. Postacím: 1388 Budapest 62., Postafiók: 86.

APRÓHIRDETÉS: Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 300 forint. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

A C-ÚJSÁG RÉGEBBI SZÁMAI megvásárolhatók az egyesület irodájában, vagy megrendelhetők utánvéttel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk a 1-76-22-57-es telefonszámon vagy levélben!

Vidéki pluszpáholy-tagjaink háromhavi tikett összegyűjtésekor igénybe vehetik a NOVOTRADE 2C Áruház csomagküldő szolgálatát.

VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:
Baja, AXIS Kft.,
Győri Bartók Béla Művelődési Ház,
Jászberényi Városi Könyvtár,
Kecskemét, SZIGMA—BIT,
Pécsi Apáczai Csere János Gimnázium,
Zalaegerszegi Ságvári Endre Gimnázium.

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:
1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57
Felelős kiadó: Horváth Judit, az egyesület elnöke
Főszerkesztő: Rados Péter, az OCE főtitkára
Felelős szerkesztő: dr. Horváth András
Művészeti szerkesztő: Bausz Sándor
Levélcím: Commodore Újság, 1388 Budapest, 62. Pf.: 86.
Index: ISSN 0237-756 X
Terjeszti a Magyar Posta
Megvásárolható a hírlapárusoknál
93.0314 MSZH Nyomda és Kiadó Kft., Budapest
Felelős vezető: Nagy László

Tagdíjfizetés

Kedves Tagtársaink!
Az év vége közeledtével felhívjuk figyelmüket az 1994. évi tagdíjuk befizetésére. Bár nem szeretnénk, de lehet, hogy kénytelenek leszünk emelni a tagdíjakat.
A tavalyihoz hasonlóan azonban az idén is lehetőséget adunk arra, hogy aki még 1993-ban befizeti a jövő évi tagdíját, az mindenképpen a régi áron teheti.
Deákpáholy 890 Ft, pluszpáholy 2050 Ft, szuperpáholy 24 000 Ft.
Régi tagjaink a csekket az újsággal együtt kapják meg, de elküldhetik a tagdíjat rózsaszínű postai utalványon is OTP számlaszámunkra.
MNB 217-98 292, OTP 565-3610-8

Országos Commodore Egyesület

C-64 C+4 AMIGA PC

tulajdonosok!

Várunk benneteket az Országos Commodore Egyesület klubdélelőttjén a Havanna Közösségi Házban minden hónap 3. vasárnapján, délelőtt 9 órától. Gépet és hosszabító hozzatok!
Cím: 1181 Budapest, Kondor Béla sétány 8.
Megközelíthető: a metró kőbánya–kispesti végállomásától a piros 136-os busszal.
Legközelebb:

január 16-án.

FELHÍVÁS

Tisztelt Tagtársunk, Olvasóink!
Kérjük, hogy ha rendelkeznek olyan működő számítógéppel, vagy jó állapotban lévő bármilyen számítógép-tartozékkal, amire nincs szükségük, akkor hozzák el egyesületi irodánkba (minden héten kedden és csütörtökön 11—15 óra között).
Mi összegyűjtjük és eljuttatjuk azokat határainkon túli (erdélyi, kárpátaljai és szlovákiai) magyar iskolákhoz.
Köszönettel:

Országos Commodore Egyesület

Tisztelt Szerkesztőség!

Pár hónapja úgy adódott, hogy egy régebben C64-esre megírt, elég terjedelmes programot IBM PC-re kellett átírnom, és ott továbbfejlesztenem. Nagyon nehezemre esett volna a közel ezer sort újra beírnom, ezért elkezdtem keresni a megoldást, hogyan vihetném át a BASIC forrásprogramot a PC-re, hogy ott aztán a QuickBASIC alatt — amely szinte teljesen kompatibilis a C64 BASIC-jével — futtathassam. Hasonló okokból másoknak is lehet ilyen problémája, ezért szeretném bemutatni az én megoldásomat.

```
100 rem *** c64-pc data transfer *
110 def fnt(x)=ti+x*60
120 d0=56577: d1=56576: d2=56589: m0=56579: m1=56578
130 poke 53280,0: poke 53281,0: print "▒▒- transfer -"
140 poke m0,128: poke m1,63: poke d0,128: a=0: f=4: gosub 630
150 wait 197,64: poke 198,0
160 rem ------
170 print "▒▒SYSTEM READY"
180 e=0: gosub 470: if e then print : gosub 700: goto 170
190 get v$: if v$="" then 180
200 if v$<>"←" then s$="hello!": gosub 230: gosub 700: poke 198,0: goto 170
210 poke m0,0: print "▒▒- end -▒": end
220 rem ====== data out (s$;) ======
230 print "sending message...": v=peek(d2): e=0
240 rem --- attention!
250 poke d0,0: a=8: f=0: gosub 630
260 rem --- ready?
270 t=fnt(10)
280 if ti>t then e=2: return
290 if (peek(d2) and 16)=0 or (peek(d0) and 127) then 280
300 rem --- message
310 if s$="" then 350
320 for i=1 to len(s$): b=asc(mid$(s$,i)): gosub 420: if e then return
330 next i
340 rem --- disconnect!
350 poke d0,128: a=0: gosub 610
360 rem --- released?
370 t=fnt(10)
380 if ti>t then e=4: return
390 if (peek(d2) and 16)=0 or (peek(d0) and 127)<>64 then 380
400 return
410 rem ====== send byte (b;)
420 gosub 610: gosub 650: if e then return
430 for x=1 to 8
440 poke d0,b: b=(b*2) and 255: gosub 610: gosub 650: if e then return
450 next x: return
460 rem ====== data in ======
470 if (peek(d2) and 16)=0 or (peek(d0) and 127)<>63 then return
480 rem ---
490 a=8: f=0: gosub 630: print "getting message...▒"
500 gosub 650: if e then return
510 if (peek(d0) and 127)=64 then e=1: return
520 gosub 610: gosub 550: if e then return
530 print chr$(b);: goto 500
540 rem ====== get byte (;b)
550 gosub 650: if e then return
560 z=peek(d0): gosub 610: b=(z and 48)*4
570 gosub 650: if e then return
580 z=peek(d0): gosub 610: b=b+(z and 63)
590 return
600 rem ====== toggle flag (a,f;)
610 f=4-f: gosub 630: return
620 rem ====== flag lines (a,f;)
630 poke d1,147 or a or f: return
640 rem ====== wait answer
650 t=fnt(2)
660 if peek(d2) and 16 then return
670 if ti>t or (peek(d0) and 64) then e=3: return
680 goto 660
690 rem ====== report & close (e;)
700 restore : for i=0 to e: read v$: next i: print "▒"v$
710 poke d0,128: poke d1,151: return
720 data "message transferred"
730 data "message received"
740 data "device not present"
750 data "lost contact"
760 data "disconnection error"
```

```
                PC printer port

  1 -STROBE 037A:0 o 0  |  14 -AUTOFD 037A:1 o 0
  2 DATA0   0378:0 o 0  |  15 ERROR   0379:3 i 1
  3 DATA1   0378:1 o 0  |  16 INIT    037A:2 o 1
  4 DATA2   0378:2 o 0  |  17 -SELIN  037A:3 o 1
  5 DATA3   0378:3 o 0  |  18 GND
  6 DATA4   0378:4 o 0  |  19 GND
  7 DATA5   0378:5 o 0  |  20 GND
  8 DATA6   0378:6 o 0  |  21 GND
  9 DATA7   0378:7 o 0  |  22 GND
 10 ACK     0379:6 i 1  |  23 GND
 11 -BUSY   0379:7 i 0  |  24 GND
 12 PAPER   0379:5 i 1  |  25 GND
 13 SELECT  0379:4 i 1  |
```



```
                C64 user port

  1 GND                       |  A GND
  2 +5V                       |  B -FLAG   DD0D:4 i  0
  3 -RESET                    |  C PB0     DD01:0 io 1
  4 CNT1                      |  D PB1     DD01:1 io 1
  5 SP1                       |  E PB2     DD01:2 io 1
  6 CNT2                      |  F PB3     DD01:3 io 1
  7 SP2                       |  H PB4     DD01:4 io 1
  8 -PC2                      |  J PB5     DD01:5 io 1
  9 -ATNOUT DD00:3 oi 0       |  K PB6     DD01:6 io 1
 10 +9VA                      |  L PB7     DD01:7 io 1
 11 +9VB                      |  M PA2     DD00:2 oi 1
 12 GND                       |  N GND
```



```
   C64                 PC

 -ATN/D1:3      -/->   SELECT/D1:4
  PA2/D1:2      -->    PAPER/D1:5
  PB7/D0:7      -->    -BUSY/D1:7

  PB6/D0:6      <--    INIT/D2:2
 -FLAG/D2:4     <--    -STROBE/D2:0
 PB0-5/D0:0-5   <==    DATA0-5/D0:0-5
```

Többféle változatot láttam a két összekapcsolására. Az egyik az volt, amikor a PC-be dugott illesztő a C64 bővítő portjával van összekapcsolva. Egy program segítségével a PC-n a C64 memóriájában lehetett közvetlenül dolgozni, sőt távirányítással programokat is lehetett futtatni. A módszer azért nem tetszett, mert a C64 ettől teljesen ,,béna'' lett, csak a PC-n keresztül lehetett hozzáférni, így pedig egy file betöltése is rendkívül körülményes tud lenni. Ráadásul a bővítő portban állandóan a Fastload kártyát tartom.

Még csábítóbb lehetőség volt az user port (esetleg a soros kábel) összekötése a PC egyik soros kommunikációs portjával, az RS—232 szabvány szerint. Sajnos amíg a PC tudomásom szerint a szokásos 12 voltos jelfeszültséget használja, addig a C64-es 5 volttal dolgozik. Az összekötés egy megfelelő (és biztosan drága) illesztő nélkül értelmetlen és veszélyes is.

Végül megtudtam, hogy a PC printer portja 5 voltos, tehát összeköthető az user porttal. Jön a következő kérdés: hogyan kell a lábakat összekötni? A mellékleten részletesen bemutatom mindkét port lábkiosztását. Ezen feltüntettem a lábhoz csatlakozó port számát, címét is és azon belül a bit sorszámát; a vonal irányát (i/o) és a gép alapállapotában a portbit értékét. Negatív állású vonal esetén persze a ,,0'' bithez tartozik a magas jelszint. A kis rajzok a csatlakozóaljzaton mutatják az első lábak helyét.

A kiosztások táblázata alatt látható az általam választott összekötési rend, ezen a portcímekben már a példaprogramokban használt jelzéseket is használtam. A legfelső vonalon a jelzés arra utal, hogy célszerűnek láttam annak megszakítását egy kapcsolóval. A C64 ugyanis a —ATN vonalat használja a lemezegység és a nyomtató megszólításakor is, és ha a PC ki van kapcsolva, akkor a bedugva tartott csatlakozókábelen ezt a vonalat fogva tarthatja. A kapcsolót az átvitel idejére kell csak zárni.

Azok számára, akik egy másféle összekötést szeretnének kipróbálni, hadd soroljam fel a tervezés általam fontosnak tartott szempontjait és néhány kiegészítő adatot a portokról. A gépek és esetleg más programok zavartalan működése érdekében a portbitek állapota lehetőleg egyezzen meg összekötött és szétkapcsolt állapotban. Az IC sérülését okozhatja, ha két kimeneti vonalat kötünk össze. A C64 CIA chipjein az irány megválasztható, de az indításkor (reset után) kapott alapállapotot kell figyelembe venni. Szintén az irányok miatt nem lehet például a C64 PB vonalait egyszerre összekötni a PC egy ki- és bemeneti portjával. Kézi kapcsolással megoldható lenne a dolog, de ezt a lehetőséget a nehézkessége és a nyilvánvaló veszélyessége miatt elvetettem.

Az user porton sajnos nagyon sok olyan láb van, amely nem csatlakozik a gép valamelyik programozható portbyte-jához, ez

```
' *** PC-C64 DATA TRANSFER *

CONST d0 = &H378
CONST d1 = &H379
CONST d2 = &H37A
DEF fnt (x) = TIMER + x

  COLOR 14, 0: CLS : PRINT "- TRANSFER -": COLOR 2
  OUT d0, 0: OUT d2, 4
  DO WHILE INKEY$ > "": LOOP: flag = 1
' -------
  DO WHILE flag < 2
    IF flag THEN PRINT : PRINT "System ready": flag = 0
    v$ = INKEY$: e = 0: GOSUB DatIn
    IF e THEN
      GOSUB Report: flag = 1
    ELSEIF v$ = CHR$(27) THEN
      flag = 2
    ELSEIF v$ > "" THEN
      s$ = "HELLO!": GOSUB DatOut: GOSUB Report
      DO WHILE INKEY$ > "": LOOP: flag = 1
    END IF
  LOOP
  COLOR 14: PRINT : PRINT "- END -": COLOR 7: END

' ====== receiving data ======
DatIn:
  IF (INP(d1) AND 176) <> 128 THEN RETURN              'signal?
' ---
  GOSUB FlagOut: f = 0                                 'ready!
  PRINT "Getting message...": COLOR 10
  DO
    GOSUB FlagIn
    IF (INP(d1) AND 144) = 16 THEN e = 1               'end of message!
    IF e = 0 THEN
      GOSUB FlagOut: b% = 0
      GOSUB ByteIn: IF e = 0 THEN PRINT CHR$(b%);
    END IF
  LOOP UNTIL e
  PRINT : OUT d0, 0: OUT d2, 5: OUT d2, 4              'disconnect
  RETURN

' === input byte (;b)
ByteIn:
  FOR x = 1 TO 8
    GOSUB FlagIn: IF e THEN RETURN
    z% = INP(d1): GOSUB FlagOut
    b% = 2 * b% + (z% AND 128) / 128
  NEXT x: b% = NOT (b%) AND 255
  RETURN

' ====== sending data ======
DatOut:
  PRINT "Sending message...": e = 0
  OUT d0, 63: OUT d2, 1: OUT d2, 0                     'attention!
  t = fnt(10)                                          'ready?
  DO UNTIL (INP(d1) AND 176) = 0
    IF TIMER > t THEN e = 2: RETURN
  LOOP: f = 0
  FOR i = 1 TO LEN(s$)                                 'message
    b% = ASC(MID$(s$, i)): GOSUB ByteOut: IF e THEN RETURN    '>>>
  NEXT i
  OUT d0, 0: OUT d2, 5: OUT d2, 4                      'disconnect!
  t = fnt(10)                                          'released?
  DO UNTIL (INP(d1) AND 176) = 48 OR e
    IF TIMER > t THEN e = 4
  LOOP
  RETURN
```

```
' === sending byte (b%;)
ByteOut:
   GOSUB FlagOut: GOSUB FlagIn: IF e THEN RETURN
   OUT d0, (b% AND 192) / 4: GOSUB FlagOut: GOSUB FlagIn: IF e THEN RETURN
   OUT d0, b% AND 63: GOSUB FlagOut: GOSUB FlagIn
   RETURN

' ====== wait answer
FlagIn:
   f = 32 - f: t = fnt(2)
   DO UNTIL (INP(d1) AND 32) = f OR e
      IF TIMER > t THEN e = 3
   LOOP: RETURN
' ====== send flag
FlagOut:
   OUT d2, 1: OUT d2, 0: RETURN
' ====== display system report & close
Report:
   RESTORE: FOR i = 0 TO e: READ v$: NEXT i: COLOR 2: PRINT v$
   OUT d0, 0: OUT d2, 4: RETURN
   DATA "message transferred"
   DATA "message received"
   DATA "device not present"
   DATA "lost contact"
   DATA "disconnection error"
```

nagyon leszűkíti a felhasználható vonalak körét. A —FLAG egy ,,lefutóél-triggerezett" vonal, ami annyit jelent, hogy a bit akkor lesz ,,1" értékű, ha a bemeneten magasról alacsonyra vált a feszültség. Ellenkező irányú változásra a vonal nem reagál, és a bit az első kiolvasással törlődik. Az user port összes megjelölt vonala közül tudomásom szerint csak a —ATN-t használja a gép önállóan, más célra is. És ez az egyetlen olyan vonal, amely visszaáll eredeti (0) értékére, ha a csatlakozást megszűntetjük.

A mellékletben látható két program, BASIC nyelven, ezeket demónak szánom az átviteli módszer tanulmányozásához. Egy billentyű leütésével üzenetet küldhetünk a másik gépnek, amelyet az kiír a képernyőre, majd mindkét gép várakozó állapotba tér vissza. A PC-s program QuickBASIC-ben készült, annak a lehető legegyszerűbb eszközeivel, elősegítve a hordozhatóságot. Külön felhívom olvasótársaim figyelmét arra, hogy ezt a programot klasszikus értelemben is strukturáltan írtam meg, egy árva GOTO nélkül. Emiatt a programok összehasonlítgatása azoknak is érdekes lehet, akik nem tervezik az összeköttetés megépítését. Néhol bizony jóval egyszerűbb a C64-es szabadabb, de azért rendezett programja.

Az átvitel ún. aszinkron típusú. Ez azt jelenti, hogy nem egy belső óra időjelzései szerint küldik a gépek a jelzéseket (akár kész a másik annak fogadására, akár nem), hanem minden mozzanathoz a másik gép nyugtázó, készenléti jelzése szükséges. Így az átvitel eredményessége teljesen független a két gép műveleti sebességétől, és ugyanolyan jól működhet (csak lassabban) BASIC, mint gépi kódú programok támogatásával. Az aszinkron módszer ugyan lassabb két jóval összehangolt gép szinkron kapcsolatánál, de annál sokkal biztonságosabb. A programok alapján, azt hiszem, bárki elkészítheti akár a vonali állapotdiagramot is, amit én most megspóroltam.

Hadd emeljek ki két finom húzást. Első pillantásra mindegynek látszik, hogy a PC—C64 irányú, kátfázisú átvitel során az adatbyte-ot a hat szabad vonalon 4—4, 5—3 vagy 6—2 arányú részekre bontva továbbítjuk. Sőt, a szabályos 4—4 arány tűnik vonzóbbnak . Ha viszont assemblyben is meg akarjuk írni a programot, akkor a 6—2 a legjobb. Ekkor ugyanis csak két léptetés kell a részek összeillesztéséhez, ami egy picivel kevesebb időbe kerül. A másik apróság az, hogy érdemes az adat kiolvasását rögtön nyugtázni, és csak utána foglalkozni az adat helyreillesztésével, feldolgozásával. Eközben ugyanis a másik gép már a következő üzenet elküldésével foglalkozhat.

Pár szót a sebességekről. Először is a kábel elkészítése után észrevettem, hogy az ACK és ERROR vonalakat is fel lehetne használni, ami a C64—PC irányban kb. kétszeres gyorsulást eredményezhetne. A bemutatott módszer szerint a BASIC-QuickBASIC változat kb. 100 Baud effektív sebességre volt képes, azaz csak az adatbyte-okat véve másodpercenként 100 bit átvitele sikerült. A Blitz fordítóval ez 3—400 Baud fölé ért. Az eltérés olyan módosításokból származik, mint az átvitel pufferezése vagy a kiírás kikapcsolása. Assemblyben már meghaladta a 3000 Baudot, ami már nagyon szép, telefonon néha profi eszközökkel sem lehet ennél jobbat elérni. Ha a PC-n is assemblyben írnám meg a fogadóprogramot, a sebesség bizonyára 10 000 fölé nőhetne. És mindez szorozható hárommal, ha a PC—C64 irányt nézzük, mert akkor 8+1 helyett csak 2+1 ütemre van szükség.

Mire használható az ilyen kapcsolat? Én például már többször vittem át programokat a másik gépre. Írtam már meg szöveget a PC egyik profi editorával, amelyet aztán a C64-es egyik programjába illesztettem. Sőt, mivel a PC-hez nincs nyomtatóm, a QuickBASIC programot is a C64-esen keresztül nyomtattam ki egy kis Seikoshával. (Arra kell csak ügyelni, hogy a lemezegység vagy nyomtató megszólítása a C64-en elrontja a —ATN állapotát.) Szabadon tervezhetjük meg az átviteli protokollt, a hibajavítással együtt. Saját parancskészlettel olvashatjuk-írhatjuk a másik gép memóriáját vagy perifériáit, méghozzá nem csak önálló programként, hanem más programok által felhasználható bővítésként is. Úgy használhatjuk az egyik gépet a másik perifériájaként, hogy közben mindkettő megtarthatja az önállóságát. Lehet, hogy a C64 elavult, de kidobni még korai lenne. Jó kísérletezést kívánok mindenkinek.

—i —a

Tisztelt Commodore Újság!

Két programot ajánlok fel maguknak bemutatásra, mert gondolom, sok C64 tulajdonosnak okoztak fejtörést azok a problémák, amelyekre a két programot készítettem.

I. Karakterező

Ez a program (mint neve is mutatja) új karakterkészlet létrehozására alkalmas. Nagy előnye, hogy BASIC-ben készült. Új karakterkészletet a régi alapján tervezhetünk, ezért nem kell az egész készletet újraírnunk, csak a régit módosítanunk. A program indítása után megkérdezi, hogy új készletet akarunk-e (15. sor). Ha újat akarunk, akkor a 10. sorban KC változóban definiált kezdőcím után írja a nagybetűs karakterkészletet, ez lesz az új készletünk. Ha nem akarunk újat, a gép egyből a módosító részre ugrik (60—140. sor). A módosítás menete: 1. Megadjuk a módosítani kívánt karakter képernyőkódját (a Felhasználói kézikönyv 126—127. oldalán található). 2. Soronként megadjuk a karakter adatait, 2-es számrendszerben. Pl. egy Ú betű:

	0	1	2	3	4	5	6	7
0. sor	0	0	0	1	1	0	0	0
1. sor	0	0	0	1	1	0	0	0
2. sor	0	0	0	0	0	0	0	0
3. sor	0	1	1	0	0	1	1	0
4. sor	0	1	1	0	0	1	1	0
5. sor	0	1	1	1	1	1	1	0
6. sor	0	0	1	1	1	1	0	0
7. sor	0	0	0	0	0	0	0	0

Ezután a gép megkérdezi, hogy akarunk-e még egyet módosítani. Ha nem, elmenti lemezre az új karakterkészletünket KC kezdőcímtől (520—600. sor). A készlet nevét az 550. sorban változtathatjuk. Ezt betöltve a készletünk aktivizálható.

```
1 REM     "┌────────────────┐"
2 REM     "|  KARAKTEREZO   |"
3 REM     "|      1993.     |"
4 REM     "|  SIDLO CSABA   |"
5 REM     "└────────────────┘"
6 REM SIDLO CSABA,3242 PARADSASVAR,ARANY J.U.33.
10 KC=14336
15 PRINT"⎕UJ KARKTERKESZLETET ? (I/N)"
20 GET A$:IF A$="" THEN 20
25 IF A$<>"I" THEN 60
30 POKE 56334,PEEK(56334)AND254:POKE 1,PEEK(1) AND 251
40 FOR I=0TO 4095:POKE KC+I,PEEK(53248+I):NEXTI
50 POKE 1,PEEK(1) OR 4:POKE 56334,PEEK(56334) OR 1
60 INPUT"⎕HANYAS KARKTER";KR
70 IF KR>255 OR KR<0 OR KR<>INT(KR) THEN 60
80 PRINT"⎕⎕          12345678⎕"
90 FOR I=0TO7
100 PRINT I;".SOR:";:INPUTA$(I):IF LEN(A$(I))<>8 THEN 100
110 GOSUB 1000:POKE KC+8*KR+I,SZ(I):NEXTI
120 PRINT"⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕MEG ?  (I/N)"
130 GET A$:IFA$="" THEN 130
140 IF A$<>"N" THEN 60
520 REM ****** KIIRO-RESZ ******
530 BC=KC+4095
550 OPEN1,8,1,"@:KARAKTERKESZLET"
560 HB=INT(KC/256):LB=KC-HB*256
570 PRINT#1,CHR$(LB);CHR$(HB)
580 FOR I=KC+1TOBC
590 PRINT#1,CHR$(PEEK(I));
600 NEXTI:CLOSE 1: END
999 REM ****** ATSZAMOLO ******
1000 FOR G=0TO7
1001 B$(D)="":C$(D)="":NEXT G
1010 FOR D=0TO7
1015 B$(D)=RIGHT$(A$(I),D+1)
1020 C$(D)=LEFT$(B$(D),1)
1030 IF C$(D)="0"THEN 1050
1040 SZ(I)=SZ(I)+2↑D
1050 NEXT D:RETURN
5000 FOR I=14336 TO 14336+4096:POKE I,0:NEXTI

READY.
```

```
5000 rem      " rCCCCCCCCCCCCCCCC¬"
5001 rem      "Bmenuvezerles sub.B"
5002 rem      "B      1993.      B"
5003 rem      "B   sidlo csaba.  B"
5004 rem      " 4CCCCCCCCCCCCCCCC┘"
5005 rem sidlo csaba,3242 Paradsasvar,arany j.u.33.
5010 a=1:sz=0:fori=1to15:ifa$(i)<>""thensz=sz+1:nexti
5020 Print"████████":for i=1tosz:Print tab(24)
5030 ifa=ithen Print"██ ";a$(i);"  █"
5040 ifa<>ithen Print"█ ";a$(i);"  "
5050 next i
5060 geta$:ifa$=""then 5060
5070 if a$="█"anda=1then Poke631,0:goto5060
5080 if a$="█"then a=a-1:goto5020
5090 if a$="█"anda=szthen Poke631,0:goto5060
5100 if a$="█"then a=a+1:goto 5020
5110 if asc(a$)=13 then fori=1tosz:a$(i)="":nexti:return
5120 goto 5060

ready.
```

```
5000 REM      " ┌────────────────┐"
5001 REM      "|MENUVEZERLES SUB.|"
5002 REM      "|     1993.       |"
5003 REM      "|  SIDLO CSABA    |"
5004 REM      " └────────────────┘"
5005 REM SIDLO CSABA,3242 PARADSASVAR,ARANY J.U.33.
5010 A=1:SZ=0:FORI=1TO15:IFA$(I)<>""THENSZ=SZ+1:NEXTI
5020 PRINT"████████":FOR I=1TOSZ:PRINT TAB(24)
5030 IFA=ITHEN PRINT"██ ";A$(I);"  █"
5040 IFA<>ITHEN PRINT"█ ";A$(I);"  "
5050 NEXT I
5060 GETA$:IFA$=""THEN 5060
5070 IF A$="]"ANDA=1THEN POKE631,0:GOTO5060
5080 IF A$="]"THEN A=A-1:GOTO5020
5090 IF A$="█"ANDA=SZTHEN POKE631,0:GOTO5060
5100 IF A$="█"THEN A=A+1:GOTO 5020
5110 IF ASC(A$)=13 THEN FORI=1TOSZ:A$(I)="":NEXTI:RETURN
5120 GOTO 5060

READY.
```

Kezdőcím	*Bekapcsolás*
2048	POKE 53272,(PEEK(53272)AND241)OR2
8192	POKE 53272,(PEEK(53272)AND241)OR8
10240	POKE 53272,(PEEK(53272)AND241)OR10
12288	POKE 53272,(PEEK(53272)AND241)OR12
14336	POKE 53272,(PEEK(53272)AND241)OR14

Én a 14336-os kezdőcímet ajánlom, mert így BASIC programunkkal együtt a karakterkészletünket is felveszi a gép. (Igaz, így is rövidebb lesz a BASIC-tár, de nem kell bajlódnunk a kezdőcímének átírásával.) Vigyázat! Ha programunkon változtatunk, a karakterkészlet eltolódik, ilyenkor újra kell töltenünk lemezről! Ha felvett készletet akarunk módosítani, a karaktermező után (!) töltsük be, és már módosítható is.

II. Menüvezérlés szubrutin

Ez a rövid rutin a menüvezérlésnek azt a fajtáját valósítja meg, amikor egymás alá kiírt menüpontok között tudunk a kurzormozgató gombokkal lépkedni. Az aktuális menüpont mindig inverz. Kiválasztani a RETURN lenyomásával tudunk. Mielőtt a szubrutinra ugranánk, el kell helyeznünk a menüpontokat A$(X) tömbváltozóban. (X = max. 15!) Innentől a rutin magától működik, egészen az első RETURN-ig. Ekkor törli A$(X) tömbváltozót, a kiválasztott menüpont sorszámát eltárolja A változóban, és visszatér a főprogramba. A menüpontok elhelyezését a képernyőn az 5020. sorban tudjuk változtatni.

Sidló Csaba

Tisztelt Szerkesztőség!

Othello:

A játékszabály: (a programban is olvasható).

Az Othellót egy 8×8-as táblán játszák. A sorokat 1—8, az oszlopokat A—H közé eső számmal, illetve betűvel jelöljük. A játék célja az, hogy minél több ellenséges figurát üssünk le. A figurák leütése úgy történik, hogy vízszintesen, függőlegesen vagy átlósan közrefogjuk az ellenfél figuráit. Ha nem tud lépni, írja be az ,,A0'' lépést. Ha fel akarja adni, a ,,Lépjek?'' kérdésre egy ,,V'' betűt.

A program stratégiája:

Egy ciklussal sorban ellenőrzi a pálya mezőit. Léphet-e oda; ha léphet, akkor hányat üt ki a lépéssel. Így kiválasztja a legjobb lépést. Ha többféle lépéssel is levehet ugyanannyit, akkor az egyforma lépések közt a véletlen (RND függvény) dönt.

30—98	Címképernyő
100—700	Zene szubrutin
710—720	Kezdeti adatok beállítása
730—750	Játékszabály
920—970; 1070—1110	Játék előtti kérdések
990—1060	Játék előtti adatok beállítása
1110—	A játék kezdete
1120—1390	A gép stratégiája
1410—1470	A gép lépése
1480—1490	Az állás kiírása
1500—1560	A játékos lépésének bekérése
1570—1970	A játékos lépésének vizsgálata
1975—1980	Az állás kiírása
1990—2010	Ellenőrzés, vége-e a játéknak
2020—2270	A játék értékelése
2280	A játék újraindítása
2300—2340	A gép lépését vizsgáló rutin (léphet-e oda)
2350—2500	A lépéseket vizsgáló rutin (hányat vesz le a lépéssel)
2510—2540	A lépést a pályára rajzoló szubrutin
2550—2750	A játéktáblát megrajzoló szubrutin
2760—3190	A karatereket átprogramozó rutin

Szűcs Gábor

```
10 POKE53280,0:POKE53281,2:POKE646,1
20 GOSUB2760
30 PRINT"□┌──┐ ┌───┐┌┐  ┌┐┌───┐┌┐   ┌┐   ┌──┐
40 PRINT"|┌─┐|└─┐┌─┘||  |||┌──┘||   ||   |┌┐|
50 PRINT"|| ||  ||  |└──┘||└─┐ ||   ||   || ||
60 PRINT"|| ||  ||  |┌──┐||┌─┘ ||   ||   || ||
70 PRINT"|└─┘|  ||  |||  ||└──┐|└──┐|└──┐|└─┘|
80 PRINT"└───┘  └┘  └┘└──┘└───┘└───┘└───┘└───┘
90 PRINT"████████████████┐└E** '*┐ACE'"
93 PRINT"▯ (C) H.NGA_| *OF|OA_E - **_C* G*BO_▯▯"
95 PRINT"              ┐┌        ┌
96 PRINT"┌┌┐  ┌┌┐┌┌┌  ┌┌┌┌┌     ┐┌┌┌"
97 PRINT"|| |  ┐┌||| ┐  |┐HH||┌    |┐┐┌"
98 PRINT"└└┘  └└└└└  └└└└└└     └└└└"
100 GOSUB110:GOTO700
110 GETA$:IFA$=" "THENRETURN
150 L1=54272:L2=54279:L3=54286
160 H1=L1+1:H2=L2+1:H3=L3+1
170 V1=L1+4:V2=L2+4:V3=L3+4
180 POKE 54296,15
190 POKE V1+1,9:POKE V2+2,0
200 GETA$:IFA$=" "THENRETURN
210 POKE V2+1,36:POKE V2+2,36
220 POKE V3+1,18:POKE V3+2,170
230 T=TI
240 GETA$:IFA$=" "THENRETURN
250 POKE V1,16:POKE V2,32:POKE V3,16
260 READ S:IF S=0 GOTO 350
270 READ X1,Y1,X2,Y2,X3,Y3
280 IF X1 THEN POKE H1,X1:POKE L1,Y1:POKE V1,17
290 IF X2 THEN POKE H2,X2:POKE L2,Y2:POKE V2,33
300 IF X3 THEN POKE H3,X3:POKE L3,Y3:POKE V3,17
310 GETA$:IFA$=" "THENRETURN
320 T=T+S
330 IF T>TI GOTO 330
340 GOTO 250
350 FORJ=L1 TO 54296:POKE J,0:NEXT J
360 RESTORE:GOTO150
370 DATA 20,34,75,21,154,8,147
380 DATA 20,34,75,25,177,0,0
390 DATA 20,38,126,28,214,6,108
400 DATA 20,43,52,25,177,0,0
410 DATA 20,34,75,21,154,8,147
420 DATA 20,43,180,25,177,0,0
430 DATA 20,38,126,22,227,8,23
440 DATA 20,0,0,25,177,0,0
450 DATA 20,34,75,21,154,8,147
460 DATA 20,34,75,25,177,0,0
470 DATA 20,38,126,28,214,6,108
480 DATA 20,43,52,25,177,0,0
490 DATA 20,34,75,21,154,8,147
500 DATA 20,0,0,25,177,0,0
510 DATA 20,32,94,22,227,8,23
520 DATA 20,0,0,19,63,6,108
530 DATA 20,34,75,21,154,8,147
540 DATA 20,34,75,25,177,0,0
550 DATA 20,38,126,21,154,7,163
560 DATA 20,43,52,17,37,0,0
570 DATA 20,45,198,28,214,7,53
580 DATA 20,43,52,34,75,0,0
590 DATA 20,38,116,28,214,0,206
600 DATA 20,34,75,22,227,0,0
610 DATA 20,32,94,25,177,6,108
620 DATA 20,25,177,22,227,0,0
630 DATA 20,28,214,21,154,7,53
640 DATA 20,32,94,19,63,8,23
650 DATA 20,34,75,21,154,8,147
660 DATA 20,0,0,25,177,6,108
670 DATA 20,34,75,21,154,4,73
680 DATA 20,0,0,0,0,0,0
```

```
690 DATA 0,-12345
700 FORJ=L1 TO 54296:POKE J,0:NEXT:CLR
710 DIMA(9,9),I4(8),J4(8),C$(8),D(2),S8(2)
720 D1=1164:S8(0)=5:S8(1)=6:S8(2)=8
730 PRINT"■■■■■A H NGA_ | ●OF| O|HELLO ┐_OG_AMJA _D>@●LI"
735 PRINT"■■■  K┐_ED A J●|┐K●●AB●L ||? (I/N)?
740 GETX$:IFX$="N"THEN920
750 IFX$<>"I"THEN740
760 PRINT"■          J●|┐K●●AB●L ■■"
770 PRINT"■■■A● O|HELLO| EG| 8●8-A● |●BL●N J●|●●●●K."
780 PRINT"■A ●O_OKA| 1-8, A● O●●LO┐OKA| A ┐● H K@●┐";
790 PRINT"■E●■ ●●●MMAL ILLEI×E BEI_■EL JEL@LJ_K."
800 PRINT"■A J●|┐K C┐LJA, HOG| MIN┐L |@BB ELEN●┐GE●";
810 PRINT"■FIG ●| _●●_NK LE. A FIG ●K LE_I┐●E ┐G|";
820 PRINT"■|@_|┐NIK, HOG| ×●●●INIE●EN F_■G■LEGE●EN";
830 PRINT"■AG| ●|LF●AN K@●_EFOGJ K A● ELLENF┐L FI-";
840 PRINT"■G ●I|. HA NEM | D L┐┐NI _JA BE A● 'A0'";
850 PRINT"■L┐┐●|. HA FEL AKA_JA ADNI A J●|┐KO|"
860 PRINT"■A 'L┐┐JEK?' K┐_D┐●_E EG| 'X' BEI_I."
900 PRINT"■          ┐_E●● '●┐ACE'":GOSUB110:S=54272:FORI=0TO24:POKES+I,0:NEXT
920 F2=0:PRINT"■■■E_■● ELLENF┐L LEG|EK? (I/N)?";:S2=0
940 GETX$
950 IFX$="N"THENPRINT" NE":GOTO980
960 IFX$<>"I"THEN940
970 PRINT" IGEN":S2=2
980 B=-1:W=1
990 D(B+1)=86:D(1)=32:D(W+1)=87
1000 I4(1)=0:I4(2)=-1:I4(3)=-1:I4(4)=-1:I4(5)=0:I4(6)=1:I4(7)=1:I4(8)=1
1010 J4(1)=1:J4(2)=1:J4(3)=0:J4(4)=-1:J4(5)=-1:J4(6)=-1:J4(7)=0:J4(8)=1
1020 C$(1)="A":C$(2)="B":C$(3)="C":C$(4)="D":C$(5)="E":C$(6)="F":C$(7)="G"
1030 C$(8)="H"
1040 FORI=0TO9:FORJ=0TO9:A(I,J)=0:NEXTJ,I
1050 A(4,4)=W:A(5,5)=W:A(4,5)=B:A(5,4)=B
1060 C1=2:H1=2:N1=4:Z=0
1070 PRINT"■■  A● \ KE●D. MI AKA_●● LENNI ●/O ?
1080 C=W:H=B
1090 GETX$
1100 IFX$="X"THEN1170
1110 IFX$<>"O"THEN1090
1120 C=B:H=W:GOTO1180
1170 GOSUB2550:GOSUB2510:GOTO1520
1180 GOSUB2550:GOTO1210
1190 IFF2=0THEN1210
1200 INPUTX$
1210 B1=-1:I3=0:J3=0:T1=C:T2=H
1220 I=1
1230 FORJ=1TO8
1240 IFA(I,J)<>0THEN1380
1250 GOSUB2300
1260 IFF1=0THEN1380
1270 U=-1:GOSUB 2350
1280 IFS1=0THEN1380
1290 IF(I-1)*(I-8)<>0THEN1310
1300 S1=S1+S2
1310 IF(J-1)*(J-8)<>0THEN1330
1320 S1=S1+S2
1330 IFS1<B1THEN1380
1340 IFS1>B1THEN1370
1360 IFRND(1)>.5THEN1380
1370 B1=S1:I3=I:J3=J
1380 NEXTJ
1390 I=I+1:IFI<9THEN1230
1400 IFB1>0THEN1440
1410 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■NEM | DOK L┐┐NI"
1420 IFZ=1THEN2020
1430 Z=1:GOTO1520
1440 ZL=0:PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■L┐┐●EM:   ";C$(J3);"-";CHR$(I3+48)
1450 I=I3:J=J3:U=1
1460 GOSUB2350
1470 C1=C1+S1+1:H1=H1-S1:N1=N1+1
```

```
1480 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]ELFOGLALIAM";S1;"ME*  "
1485 PRINT"Q]]]]]]]]]]]]]]]]]]]]]]]]]]]]]QQQQQQQQQQQQEN IT  ";C1"  ME*  "
1487 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]I TD   ";H1"  ME*  "
1490 GOSUB2510
1500 IFH1=0THEN2020
1510 IFN1=64THEN2020
1520 T1=H:T2=C
1530 PRINT"QQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]HOX* L  **?          ";
1540 POKE198,0:WAIT198,1:GETZ1$:PRINTZ1$;"-";:POKE198,0:WAIT198,1:GETZ2$:PRINTZ2
$
1550 Z1=ASC(Z1$):Z2=ASC(Z2$)
1560 IFZ1>47ANDZ1<57ANDZ2>64ANDZ2<73ORZ1>64ANDZ1<73ANDZ2>47ANDZ2<57THEN1580
1570 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]HIBA! \_D BE  J_A!"
1580 IFZ1>57THEN1600
1590 I=Z1-48:X$=CHR$(Z2):GOTO1610
1600 I=Z2-48:X$=CHR$(Z1)
1610 IFI<0THEN1530
1620 IFI>8THEN1530
1630 IFI<>0THEN1720
1640 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]L  JEK?             ":PRINT"
1650 GETX$
1660 IFX$="V"THEN2020
1670 IFX$="N"THEN1530
1680 IFX$<>"I"THEN1650
1690 PRINT"II]]]]]]]]]]]]]]]]]]]]]]]]]]]]                 "
1700 IFZ=1THEN2020
1710 Z=1:GOTO1190
1720 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       "
1730 FORJ=1TO8
1740 IFC$(J)=LEFT$(X$,1)THEN1770
1750 NEXTJ
1760 GOTO950
1770 IFA(I,J)=0THEN1810
1780 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]] HIBA !"
1790 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]\_D BE  J_A!"
1800 GOTO1530
1810 GOSUB2300
1820 IFF1=1THEN1860
1830 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]] HIBA !"
1840 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]\_D BE  J_A!"
1850 GOTO1530
1860 U=-1
1870 GOSUB2350
1880 IFS1>0THEN1920
1890 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]           "
1900 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]\_D BE  J_A ! "
1910 GOTO1530
1920 Z=0
1930 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]Q                    "
1940 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]                 "
1950 PRINT"QQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]ELFOGLALI*L";S1;"ME*  "
1960 U=1:GOSUB2350
1970 H1=H1+S1+1:C1=C1-S1:N1=N1+1
1975 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]EN IT  "C1"  ME*  "
1977 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]I TD   "H1"  ME*  "
1980 GOSUB 2510
1990 IFC1=0THEN2020
2000 IFN1=64THEN2020
2010 GOTO1190
2020 PRINT"QQQQQQQQQQQQQ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]QQNEKED                ";H1
2030 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]NTG I*EIED XAN NEKEM
2040 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]PEDIG                  ";C1:PRINT"
2050 IFH1=C1THEN2090
2060 IFH1>C1THEN2110
2070 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]G  *IEM !!!           X]]]]]]]]]]]]]]]]]]]]]]]]]]]]
2080 GOTO2130
2090 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]DONTEILEN !!!          X]]]]]]]]]]]]]]]]]]]]]]]]]]]]
2100 GOTO2280
2110 PRINT"]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]G  *I L  !!!           X]]]]]]]]]]]]]]]]]]]]]]]]]]]]
2120 C9=C1:C1=C1-H1
```

```
2130 IFC1>0THEN2150
2140 C1=-C1
2150 C1=(64*C1)/N1
2160 IFC1<11THEN2260
2170 IFC1<25THEN2250
2180 IFC1<39THEN2240
2190 IFC1<53THEN2220
2200 IFC9>H1THEN2260
2210 PRINT"[illegible] **ENI*LI* >AG I.":PRINT"G_AI ∠*LOK":GOTO2280
2220 IFC9>H1THEN2250
2230 PRINT"[illegible]NEH‾* C*AI A >OLI":GOTO2280
2240 PRINT"[illegible]M‾G G IAKO_OLNOD KELL":GOTO2280
2250 PRINT"[illegible]ELMEG I.(**┌D*‾AL>":GOTO2280
2260 IFC9<H1THEN2220
2270 PRINT"[illegible]I ELJE*EN IEHEI*‾GIELEN >AG I!"
2280 PRINT"[illegible]PRE** *PACE":GOSUB110:CLR:RUN30
2290 GOTO2280
2300 FORI1=-1TO1
2310 FORJ1=-1TO1
2320 IFA(I+I1,J+J1)=T2THEN2340
2330 NEXTJ1,I1:F1=0:RETURN
2340 F1=1:RETURN
2350 S1=0
2360 FORK=1TO8
2370 I5=I4(K):J5=J4(K):I6=I+I5:J6=J+J5:S3=0
2380 IFA(I6,J6)<>T2THEN2490
2390 S3=S3+1:I6=I6+I5:J6=J6+J5
2400 IFA(I6,J6)=T1THEN2430
2410 IFA(I6,J6)=0THEN2490
2420 GOTO2390
2430 S1=S1+S3
2440 IFU<>1THEN2490
2450 I6=I:J6=J
2460 FORK1=0TOS3
2470 A(I6,J6)=T1:I6=I6+I5:J6=J6+J5
2480 NEXTK1
2490 NEXTK
2500 RETURN
2510 FORI=1TO8:FORJ=1TO8
2520 POKE646,7:XY=D(A(I,J)+1):IFXY=86THENXY=77:POKE646,7
2522 IFXY=87THENXY=78:POKE646,7
2525 POKED1+80*I+2*J-20,XY
2540 POKE646,1:NEXTJ,I:RETURN
2550 PRINT"[illegible]        --- O I H E L L O ---
2560 PRINT"[illegible]  A B C D E F G H
2570 PRINT" ┌┬┬┬┬┬┬┬┐
2580 PRINT"1| | | | | | | | |1"
2590 PRINT" ├┼┼┼┼┼┼┼┤
2600 PRINT"2| | | | | | | | |2"
2610 PRINT" ├┼┼┼┼┼┼┼┤
2620 PRINT"3| | | | | | | | |3"
2630 PRINT" ├┼┼┼┼┼┼┼┤
2640 PRINT"4| | | | | | | | |4"
2650 PRINT" ├┼┼┼┼┼┼┼┤
2660 PRINT"5| | | | | | | | |5"
2670 PRINT" ├┼┼┼┼┼┼┼┤
2680 PRINT"6| | | | | | | | |6"
2690 PRINT" ├┼┼┼┼┼┼┼┤
2700 PRINT"7| | | | | | | | |7"
2710 PRINT" ├┼┼┼┼┼┼┼┤
2720 PRINT"8| | | | | | | | |8"
2730 PRINT" └┴┴┴┴┴┴┴┘
2740 PRINT"  A B C D E F G H
2750 RETURN
2760 PRINTCHR$(142):PRINT"[illegible]       KERLEK VARJ!"
2770 POKE52,48:POKE56,49
2780 POKE56334,PEEK(56334) AND 254
2790 POKE1,PEEK(1)AND251
2800 FORI=0TO511:POKE12288+I,PEEK(I+53248):NEXT
2810 POKE 1,PEEK(1)OR4
```

```
2820 POKE56334,PEEK(56334)OR1
2830 POKE53272,(PEEK(53272)AND 240)+12
2840 RESTORE
2850 READA:IFA<>-12345THEN2850
2860 FORI=12808TO12808+7:READA:POKEI,A:NEXT
2870 FORI=12840TO12840+7:READA:POKEI,A:NEXT
2880 DATA 6,28,60,102,126,102,102,0,12,8,126,96,120,96,126,0
2890 FORI=12872TO12872+7:READA:POKEI,A:NEXT
2900 FORI=12920TO12920+7:READA:POKEI,A:NEXT
2910 DATA12,8,60,24,24,24,60,0,12,8,60,102,102,102,60,0
2920 FORI=12288TO12288+7:READA:POKEI,A:NEXT
2930 FORI=13128TO13128+7:READA:POKEI,A:NEXT
2940 DATA36,0,60,102,102,102,60,0,54,36,60,102,102,102,60,0
2945 FORI=12416TO12496+8:POKEI+512,PEEK(I):NEXT
2950 FORI=13248TO13248+7:READA:POKEI,A:NEXT
2960 FORI=12512TO12512+7:READA:POKEI,A:NEXT
2970 DATA24,16,102,102,102,102,60,0,36,0,102,102,102,102,60,0
2980 FORI=13264TO13264+7:READA:POKEI,A:NEXT
2990 FORI=13184TO13184+7:READA:POKEI,A:NEXT
3000 DATA 54,36,102,102,102,102,60,0,0,0,63,32,47,40,40,40
3010 FORI=12816TO12816+7:READA:POKEI,A:NEXT
3020 FORI=12824TO12824+7:READA:POKEI,A:NEXT
3030 DATA40,40,40,40,40,40,40,40,0,0,255,0,255,0,0,0
3040 FORI=13192TO13192+7:READA:POKEI,A:NEXT
3050 FORI=13016TO13016+7:READA:POKEI,A:NEXT
3060 DATA40,40,239,0,255,0,0,0,40,40,239,0,239,40,40,40
3070 FORI=13200TO13200+7:READA:POKEI,A:NEXT
3080 FORI=13160TO13160+7:READA:POKEI,A:NEXT
3090 DATA0,0,255,0,239,40,40,40,40,40,47,32,63,0,0,0
3100 FORI=13288TO13288+7:READA:POKEI,A:NEXT
3110 FORI=13168TO13168+7:READA:POKEI,A:NEXT
3120 DATA40,40,232,8,248,0,0,0,0,0,248,8,232,40,40,40
3130 FORI=13144TO13144+7:READA:POKEI,A:NEXT
3140 FORI=13208TO13208+7:READA:POKEI,A:NEXT
3150 DATA40,40,47,32,47,40,40,40,40,40,232,8,232,40,40,40
3160 FORI=12904TO12904+7:READA:POKEI,A:NEXT
3170 FORI=12912TO12912+7:READA:POKEI,A:NEXT
3180 DATA195,102,60,24,60,102,195,0,24,60,102,195,195,102,60,24,0,0
3185 PRINT"]
3190 RETURN

READY.
```

Tisztelt Szerkesztőség!

Ez a program egy 3×8×21-es sprite elkészítését teszi könyebbé. A program elindítása után a képernyőn megjelennek a főbb tudnivalók. A betűköz-billentyű megnyomásával a programot elindíthatjuk. Ekkor megjelenik a képernyőn a 3×8×21-es sprite ábrája, amibe elkezdhetjük megtervezni az ábránkat. Itt tudjuk használni a Commodore billentyűzetéből az I, J, K, L, M billentyűket vagy a joystick 2. portját.

Ha kész az ábránk, nyomjuk meg az Fl billentyűt, és a gép kiszámolja a sprite értékét. A programból kilépni az F3 billentyűvel lehet. A program újraindítását az F5 billentyűvel érjük el. Az ábra törlését az F7 billentyű végzi.

Ha a gép kiszámolta a számokat, akkor választhatunk a képernyőn megjelenő lehetőségek közül. (Választás csak joystickkal 2. port.) Kérhetjük a grafika megmutatását, számokat, kiléphetünk, majd visszatérhetünk programunk elejére. (A grafikát nagyítani, illetve kicsinyíteni egyaránt lehet.)

Tisztelettel **Vígh Mihály**

```
1 REM ***************************
2 REM *          [64]            *
3 REM *      VIGH MIHALY         *
4 REM *        TURKEVE           *
5 REM *    SPRITE KESZITO        *
6 REM ***************************
10 PRINT"]":REM * KEPERNYO TORLES *
12 POKE53281,6:POKE53280,6:REM *
   KERET ES HATER SZIN *
18 PRINT"M -MOZGATAS:"
20 PRINT" FOL.            I "
30 PRINT" BAL/JOBB.    J     L "
32 PRINT" LE.             K"
34 PRINT" KOCKA MEGJELOLES [M]"
35 PRINT" VAGY JOYSTICK"
36 PRINT" F1 GRAFIKAI PONTOK
   KISZAMITASA"
37 PRINT" F3 KILEPES A PROGRAMBOL"
38 PRINT" F5 PROGRAM UJRA INDITASA"
39 PRINT" F7 TORLES"
40 PRINT"M-SPACE"
```

```
41 GETA$:IFA$=" "THENPRINT"□":GOTO44
42 GOTO41
44 X=1111:DIMN(530):DIMW(70):PRINT"□"
45 POKEX,42:REM * KEZDO PONT HEJE *
46 REM * KERET MEGRAJZOLASA *
50 FORI=1070TO1095:POKEI,160:NEXT
60 FORI=1950TO1975:POKEI,160:NEXT
70 FORI=1110TO1910STEP40:POKEI,160:NEXT
80 FORI=1135TO1944STEP40:POKEI,160:NEXT
90 REM * IRANYITAS BILLENTYU *
100 GETA$:IFA$="I"THENZ=X:X=X-40:GOTO250
120 IFA$="K"THENZ=X:X=X+40:GOTO250
130 IFA$="L"THENZ=X:X=X+1 :GOTO250
140 IFA$="J"THENZ=X:X=X-1 :GOTO250
150 IFA$="M"THEN310
151 REM * KEZELO BILLENTYUK *
152 IFA$=CHR$(133)THEN400
153 IFA$=CHR$(136)THENRUN44
154 IFA$=CHR$(134)THENPRINT"□":END
156 IFA$=CHR$(135)THENRUN
157 REM * IRANYITAS JOYSTICK [2] *
160 J=PEEK(56320)
170 IF(JAND1)=0THENZ=X:X=X-40:GOTO250
180 IF(JAND2)=0THENZ=X:X=X+40:GOTO250
190 IF(JAND4)=0THENZ=X:X=X-1 :GOTO250
200 IF(JAND8)=0THENZ=X:X=X+1 :GOTO250
210 IF(JAND16)=0THEN310
220 GOTO100
230 REM * CSILLAG HEJZET ELLENORZESE *
250 IFX<1111THENX=Z:GOTO100
260 IFX>1934THENX=Z:GOTO100
270 FORI=1110TO1910STEP40:IFX=ITHENX=Z:GOTO100
275 NEXT
280 FORI=1135TO1944STEP40:IFX=ITHENX=Z:GOTO100
285 NEXT
289 REM * CSILLAG ATHEJEZESE *
290 IFPEEK(Z)=102THENPOKEX,42:GOTO100
300 POKEX,42:POKEZ,32:GOTO100
310 POKEX,102:GOTO100
390 REM ***********************************
400 REM *** SZAMOLAS ***
401 PP=1111:YY=1935:L=L-1
405 FORXX=PPTOYYSTEP40
410 FORO=XXTOXX+24:L=L+1:IFPEEK(O)=102THENN(L)=1:GOTO 412
411 N(L)=0
412 PRINT"▒▒▒▒";L:NEXT:L=L-1:NEXTXX
650 FORI=1TO504STEP8:C=0:E=E+1:Q=I-1
660 IFN(Q)=1THENC=C+128
670 IFN(Q+1)=1THENC=C+64
680 IFN(Q+2)=1THENC=C+32
690 IFN(Q+3)=1THENC=C+16
700 IFN(Q+4)=1THENC=C+8
710 IFN(Q+5)=1THENC=C+4
720 IFN(Q+6)=1THENC=C+2
730 IFN(Q+7)=1THENC=C+1
740 W(E)=C:PRINT"▒▒▒▒▒";E:NEXT
750 PRINT"▒▒SPACE▒"
760 GETA$:IFA$=" "THEN780
770 GOTO760
775 REM ***********************************
780 PRINT"□▒"
```

```
790 PRINT"▒╭─────┬──────────────────────────╮"
800 PRINT"▒|PLAY| GRAFIKA MEGMUTATAS  |"
810 PRINT"▒╰─────┴──────────────────────────╯▒"
820 PRINT"▒╭─────┬──────────────────────────╮"
830 PRINT"▒|LIST| SZAMOK KIKERESE      |"
840 PRINT"▒╰─────┴──────────────────────────╯▒"
850 PRINT"▒╭─────┬──────────────────────────╮"
860 PRINT"▒|END | KILEPES              |"
870 PRINT"▒╰─────┴──────────────────────────╯▒"
880 PRINT"▒╭─────┬──────────────────────────╮"
890 PRINT"▒|RUN | VISSZA AZ ELEJERE    |"
900 PRINT"▒╰─────┴──────────────────────────╯▒"
908 Y=1906:E=32:POKEY,30
909 REM * JOYSTIC [2] IRANYITAS *
910 J=PEEK(56320)
920 IF(JAND1)=0THENZ=Y:Y=Y-40:GOTO980
930 IF(JAND2)=0THENZ=Y:Y=Y+40:GOTO980
940 IF(JAND4)=0THENZ=Y:Y=Y-1 :GOTO980
950 IF(JAND8)=0THENZ=Y:Y=Y+1 :GOTO980
960 IF(JAND16)=0THEN1050
970 GOTO910
980 IFY<1024THENY=Z:GOTO910
990 IFY>2023THENY=Z:GOTO910
1010 POKEZ,E:E=PEEK(Y):POKEY,30:GOTO910
1020 REM * NYIL HEJENEK ELLENORZESE *
1050 FORI=1105TO1108:IFY=ITHEN1100
1055 NEXT
1060 FORI=1265TO1268:IFY=ITHEN1400
1065 NEXT
1070 FORI=1425TO1428:IFY=ITHENPRINT"▒":END
1075 NEXT
1080 FORI=1585TO1588:IFY=ITHENRUN
1085 NEXT
1090 GOTO 910
1100 PRINT"▒":REM *** GRAFIKA ***
1110 V=53248:RESTORE
1120 POKE2042,13:POKEV+21,4
1130 FORL=1TO63:POKE831+L,W(L):NEXT:REM * ADATOK BETOLTESE *
1140 POKEV+4,150:POKEV+5,150:REM * GRAFIKA ELHEJEZESE *
1150 PRINT"▒▒▒╭─────┬──────────────┬─────╮"
1155 PRINT"▒| [K] | NAGYITAS | [N] |"
1160 PRINT"▒╰─────┴──────────────┴─────╯"
1165 PRINT"▒▒SPACE▒":K=1
1170 GETA$:IFA$=""THEN1170
1180 IFA$="N"THEN1210
1190 IFA$=" "THENPOKEV+4,0:POKEV+5,0:GOTO780
1195 IFA$="K"THEN1220
1200 GOTO1170
1210 POKE53248+23,4:POKE53248+29,4:REM * NAGYITAS *
1215 GOTO1170
1220 POKE53248+23,3:POKE53248+29,3:REM KICSINYITES *
1225 GOTO1170
1399 REM ******************************
1400 PRINT"▒":REM * SZAMOK KIIRATASA *
1410 FORI=1TO63STEP3
1420 PRINT,W(I),W(I+1),W(I+2):NEXT
1435 PRINT"▒SPACE▒"
1440 GETA$:IFA$=" "THEN780
1445 GOTO1440

READY.
```

```
100 REM ##############################
110 REM #          "                 #
120 REM #        A M O B A           #
130 REM #                            #
131 REM #      COMMODORE 64          #
140 REM #                            #
150 REM #  KESZITETTE : GABRYEL      #
160 REM #                            #
170 REM ##############################
180 :
190 POKE53280,.:POKE53281,.:POKE646,12
200 PRINT"[clr]"TAB(10)"
210 PRINTTAB(10)"KARAKTEREK KESZITESE !
220 :
230 REM ****************
240 REM * CHAR ATIRO *
250 REM ****************
260 :
270 POKE 56334,PEEK(56334) AND 254
280 POKE 1,PEEK(1) AND 251
290 FOR I=0 TO 1024:POKE12288+I,PEEK(53248+I):NEXT
300 POKE 1,PEEK(1) OR 4
310 POKE 56334,PEEK(56334) OR 1
320 :
330 FOR Q=1 TO 64:S=0
340 FOR I=0 TO 3:A=PEEK(12288+8*Q+I)
350 FOR U=0 TO 1
360 POKE 12288+8*(Q+128)+S,A:S=S+1
370 NEXT
380 NEXT
390 S=0
400 FOR I=4 TO 7:A=PEEK(12288+8*Q+I)
410 FOR U=0 TO 1
420 POKE 12288+8*(Q+192)+S,A:S=S+1
430 NEXT
440 NEXT
450 NEXT
460 POKE53272,28
470 FORI=0TO7:POKE12288+8*160+I,255:
    NEXT:Q=0:U=0:S=0:
480 :
481 PRINT"[clr]VEDEKEZO JATEKSTILUST ?"
482 GETA$:IFA$=""THEN482
483 POKE2,0
484 IF A$="I" THEN POKE 2,26
490 REM *************
500 REM * KEZDOKEP *
510 REM *************
520 :
530 PRINT"[clr]"
540 PRINTTAB(20)"[ctrl]"
550 PRINTTAB(15)"[ctrl]"
560 PRINTTAB(15)"[ctrl]"
570 PRINTTAB(15)"[ctrl]"
580 PRINTTAB(14)"KESZITETTE :"
590 PRINTTAB(16)"[ctrl]GABRYEL"
600 PRINTTAB(16)"[ctrl]"
601 PRINTTAB(22)"[ctrl]
610 PRINTTAB(14)"1 - SZABALY"
611 PRINTTAB(20)"[ctrl]
620 PRINTTAB(14)"2 - JATEK"
630 FORI=1 TO 39:POKE1024+I,64
640 POKE1984+I,64:NEXT
650 FOR I=1 TO 23:POKE 1024+40*I,66
```

Tisztelt Szerkesztőség!

Az AMŐBA nevű program a meghirdetett pályázatra készült. Ezt a fajta amőbát háromszor-hármas táblán játszák. Mi a kör alakú bábuval vagyunk, a célunk, hogy egy sorba vagy oszlopba kirakjuk a három bábut.

A gép lépésének vizsgálatát úgy oldottam meg, hogy a vizsgálat elején elugrik egy szubrutinra, ahol megvizsgálja a 2. tárcímet. (Itt tároltam, hogy milyen játékstílust választottunk.) Ha a cím egyenlő nullával, visszaugrik a szubrutin elejére. Ha viszont a cím nem nulla, akkor megvizsgálja, hogy egy sorban vagy oszlopban van-e neki 2 bábuja. Ha van, akkor a harmadikat közéjük teszi és elugrik a vizsgálat végére, ahol kirakja a bábut.

Ha nem talál egy sorban két bábut, akkor visszaugrik a vizsgálatok elejére. Ott a vizsgálatok elején megnézi, hogy van-e nekünk egy sorban két bábunk. Ha van, akkor az övét közéjük teszi. Ha nincs, akkor egy véletlenszám dönti el, hogy hova teszi a maga bábuját.

A program felületesen

230—480	karakterek készítése
481—484	beállítja a játékstílust (2. tárcím)
490—690	kezdőkép

```
660 POKE 1063+40*I,66:NEXT
670 POKE1024,112:POKE1063,110
680 POKE1984,109:POKE2023,125
690 :
700 REM ********************
710 REM * VALASZTAS(J,S) *
720 REM ********************
730 :
740 GETA$:IFA$=""THEN740
750 IF A$="1" THEN 830
760 IF A$="2" THEN 1050
770 GOTO 740
780 :
790 REM ************
800 REM * SZABALY *
810 REM ************
820 :
830 PRINT"[illegible]"
840 PRINTTAB(15)"      ,"
850 PRINTTAB(15)"[illegible]SZABALY"
860 PRINTTAB(15)"[illegible]"
870 FORI=0TO39:POKE1264+I,45:NEXT
880 PRINT"[illegible]"
890 PRINT"          , ,"
900 PRINT"  A JATEKOT BISZTOSAN MINDENKI ISMERI.
910 PRINT"                 — ,       ,  ,
920 PRINT" EZT A FAJTA AMOBAT 3*3-AS TABLAN
930 PRINT"  ;    ,     , ,   ,        ,
940 PRINT" JATSZAK.A JATEK LENYEGE, HOGY EGYMAS
950 PRINT"  ,                   ,
960 PRINT" MELLE KIRAKJUNK 3 EGYFORMA BABUT.
970 PRINT"              —         ,
980 PRINT" AZ NYER AKI ELOBB KIRAKJA A 3 BABUT."
990 PRINT"         ,    ,      ,
1000 PRINT" JO SZORAKOZAST !
1010 :
1020 POKE198,0:WAIT198,1
1030 :
1040 REM ************
1050 REM * JATEKKEP *
1060 REM ************
1070 :
1080 :
1090 :
1100 PRINT"[illegible]"
1110 PRINTTAB(12)"[illegible]"
1120 FOR I=1 TO 3
1130 PRINTTAB(12)"[illegible]"
1140 PRINTTAB(12)"[illegible]"
1150 PRINTTAB(12)"[illegible]"
1160 PRINTTAB(12)"[illegible]"
1170 PRINTTAB(12)"[illegible]"
1180 NEXT
1181 PRINTTAB(12)"[illegible]"
1182 PRINTTAB(12)"[illegible]"
1183 PRINT"[illegible]"TAB(10)"[illegible]"
1190 :
1200 REM *********
1210 REM * JATEK *
1220 REM *********
1230 :
1240 DIM A(3,3),X(3),Y(3)
1250 FOR I=1 TO 3:READ X(I),Y(I):NEXT
1260 DATA 14,3,19,8,24,13
```

700—780	a lenyomott billentyű kiértékelése
790—1030	szabály kiírása
1040—1190	kirajzolja a képet
1200—1280	beállítja a változókat
1290—1380	X rakó szubrutin
1390—1490	O rakó szubrutin
1500—1630	a játékos lépésének érvényességi vizsgálata
1640—1790	nyerési vizsgálat (a játékos megnyerte-e a játszmát?)
1800—2180	a gép lépésének vizsgálata
2190—2290	nyerési vizsgálat (a gép megnyerte-e a játszmát?)
3010—4090	játékvégi képek
4100—4310	a gép lépésének vizsgálata a nem védekező játékstílusban

Fekete Gábor

```
1270 GOTO 1490
1280 :
1290 REM ********************
1300 REM * X RAKO SZUBRUTIN *
1310 REM ********************
1320 :
1330 POKE1024+40*Y(Y1)+X(X1),77
1340 POKE1024+40*Y(Y1)+X(X1)+1,78
1350 POKE1024+40*(Y(Y1)+1)+X(X1),78
1360 POKE1024+40*(Y(Y1)+1)+X(X1)+1,77
1361 PRINT"▒"TAB(30)"▒▒▒OSZLOP:"X1:PRINT"▒"TAB(30)"▒▒▒▒SOR:"Y1
1370 RETURN
1380 :
1390 REM ********************
1400 REM * O RAKO SZUBRUTIN *
1410 REM ********************
1420 :
1430 POKE1024+40*Y(Y0)+X(X0),85
1440 POKE1024+40*Y(Y0)+X(X0)+1,73
1450 POKE1024+40*(Y(Y0)+1)+X(X0),74
1460 POKE1024+40*(Y(Y0)+1)+X(X0)+1,75
1461 PRINT"▒▒▒▒OSZLOP:"X0"▒▒▒▒▒▒SOR:"Y0
1470 RETURN
1480 :
1490 :
1500 GETA$:IFA$=""THEN1500
1510 PRINT"▒                              "
1520 P=ASC(A$)-64
1530 IF P<1 OR P>3 THEN PRINT"▒NINCS ILYEN SOR !":GOTO1500
1540 X0=P:PRINT"▒"A$
1550 GET A$:IFA$=""THEN1550
1560 PRINT"▒                              "
1570 P=ASC(A$)-48
1580 IF P<1 OR P>3 THEN PRINT"▒NINCS ILYEN OSZLOP !":GOTO1550
1590 Y0=P
1600 IF A(X0,Y0)<>0 THEN PRINT"▒ ODA MAR LEPTEK !":GOTO 1500
1610 GOSUB 1400:A(X0,Y0)=1
1620 :
1630 :
1640 REM **********************
1650 REM * NYERESI VIZSGALAT *
1670 REM **********************
1680 :
1690 FOR I=1 TO 3
1700 IF A(1,I)=1 AND A(2,I)=1 AND A(3,I)=1 THEN 4010
1710 NEXT
1720 FOR I=1 TO 3
1730 IF A(I,1)=1 AND A(I,2)=1 AND A(I,3)=1 THEN 4010
1740 NEXT
1750 IF A(1,1)=1 AND A(2,2)=1 AND A(3,3)=1 THEN 4010
1760 IF A(3,1)=1 AND A(2,2)=1 AND A(1,3)=1 THEN 4010
1780 Q=0:W=0:XX=0:YY=0
1790 :
1800 REM ****************
1810 REM * A GEP LEPESE *
1820 REM ****************
1830 GOTO 4100
1840 Q=Q+1
1850 IF A(1,Q)=1 AND A(2,Q)=1 AND A(3,Q)=0 THEN XX=3:YY=Q:GOTO 2150
1860 IF A(1,Q)=1 AND A(3,Q)=1 AND A(2,Q)=0 THEN XX=2:YY=Q:GOTO 2150
1870 IF A(2,Q)=1 AND A(3,Q)=1 AND A(1,Q)=0 THEN XX=1:YY=Q:GOTO 2150
1880 IF Q<3 THEN 1840
1890 :
1900 Q=0
```

```
1910 Q=Q+1
1920 IF A(Q,1)=1 AND A(Q,2)=1 AND A(Q,3)=0 THEN XX=Q:YY=3:GOTO 2150
1930 IF A(Q,1)=1 AND A(Q,3)=1 AND A(Q,2)=0 THEN XX=Q:YY=2:GOTO 2150
1940 IF A(Q,2)=1 AND A(Q,3)=1 AND A(Q,1)=0 THEN XX=Q:YY=1:GOTO 2150
1950 IF Q<3 THEN 1910
1960 :
1970 IF A(1,1)=1 AND A(2,2)=1 AND A(3,3)=0 THEN XX=3:YY=3:GOTO 2150
1980 IF A(1,1)=1 AND A(3,3)=1 AND A(2,2)=0 THEN XX=2:YY=2:GOTO 2150
1990 IF A(2,2)=1 AND A(3,3)=1 AND A(1,1)=0 THEN XX=1:YY=1:GOTO 2150
2000 :
2010 IF A(3,1)=1 AND A(2,2)=1 AND A(1,3)=0 THEN XX=1:YY=3:GOTO 2150
2020 IF A(3,1)=1 AND A(1,3)=1 AND A(2,2)=0 THEN XX=2:YY=2:GOTO 2150
2030 IF A(2,2)=1 AND A(1,3)=1 AND A(3,1)=0 THEN XX=3:YY=1:GOTO 2150
2040 :
2050 FOR I=1 TO 3
2060 FOR U=1 TO 3
2070 IF A(U,I)=0 THEN U=13
2080 NEXT:NEXT
2090 :
2100 IF U<>13 THEN RUN500
2110 :
2120 XX=INT(RND(1)*3+1)
2130 YY=INT(RND(1)*3+1)
2140 IF A(XX,YY)<>0 THEN 2120
2150 A(XX,YY)=2
2160 X1=XX:Y1=YY
2170 GOSUB 1300
2180 :
2190 REM ************
2200 REM * NYERTES ?*
2210 REM ************
2220 :
2230 FOR I=1 TO 3
2240 IF A(1,I)=2 AND A(2,I)=2 AND A(3,I)=2 THEN Z=13
2250 IF A(I,1)=2 AND A(I,2)=2 AND A(I,3)=2 THEN Z=13
2260 NEXT
2270 IF A(1,1)=2 AND A(2,2)=2 AND A(3,3)=2 THEN Z=13
2280 IF A(1,3)=2 AND A(2,2)=2 AND A(3,1)=2 THEN Z=13
2290 IF Z=13 THEN 3010
3000 GOTO 1500
3010 :
3020 PRINT"▓"
3030 PRINT"▓▓▓▓▓▓▓▓▓▓        ,    —                ,"
3040 PRINT"▓▓▓▓A GEP UGYESEBB NALAD !"
3050 PRINT"▓▓▓▓▓▓▓▓▓▓▓ ,                ,"
3060 PRINT"▓▓▓▓JATSZUK MEG ? (I/N)"
3070 GETA$:IFA$=""THEN3080
3080 IF A$="I" THEN RUN500
3090 IF A$="N" THEN END
4000 GOTO 3070
4010 PRINT"▓"
4020 PRINT"▓▓▓▓▓▓▓▓▓▓               ,"
4030 PRINT"▓▓▓▓NYERTEL !"
4040 PRINT"▓▓▓▓▓▓▓▓▓▓ ,              ,"
4050 PRINT"▓▓▓▓JATSZUNK MEG ? (I/N)"
4060 GETA$:IFA$=""THEN4060
4070 IF A$="I" THEN RUN500
4080 IF A$="N" THEN END
4090 GOTO 4060
4100 IFPEEK(2)=26 THEN GOTO 1940
4101 I=0
4110 I=I+1
4120 IFA(1,I)=2ANDA(2,I)=2ANDA(3,I)=0THENXX=3:YY=I:GOTO 2150
```

```
4130 IFA(1,I)=2ANDA(2,I)=0ANDA(3,I)=2THENXX=2:YY=I:GOTO 2150
4140 IFA(1,I)=0ANDA(2,I)=2ANDA(3,I)=2THENXX=1:YY=I:GOTO 2150
4150 IF I<3 THEN 4110
4160 :
4170 I=0
4180 I=I+1
4190 IF A(I,1)=2 AND A(I,2)=2 AND A(I,3)=0 THEN XX=I:YY=3:GOTO 2150
4200 IF A(I,1)=2 AND A(I,2)=0 AND A(I,3)=2 THEN XX=I:YY=2:GOTO 2150
4210 IF A(I,1)=0 AND A(I,2)=2 AND A(I,3)=2 THEN XX=I:YY=1:GOTO 2150
4220 IF I<3 THEN 4180
4230 :
4240 IF A(1,1)=2 AND A(2,2)=2 AND A(3,3)=0 THEN XX=3:YY=3:GOTO 2150
4250 IF A(1,1)=2 AND A(2,2)=0 AND A(3,3)=2 THEN XX=2:YY=2:GOTO 2150
4260 IF A(1,1)=0 AND A(2,2)=2 AND A(3,3)=2 THEN XX=1:YY=1:GOTO 2150
4270 :
4280 IF A(3,1)=2 AND A(2,2)=2 AND A(1,3)=0 THEN XX=1:YY=3:GOTO 2150
4290 IF A(3,1)=2 AND A(2,2)=0 AND A(1,3)=2 THEN XX=2:YY=2:GOTO 2150
4300 IF A(3,1)=0 AND A(2,2)=2 AND A(1,3)=2 THEN XX=3:YY=1:GOTO 2150
4310 GOTO 1840

READY.
```

Tippek–trükkök nyomtatóhoz

Starpainter – modullal

Akinek Centronics-csatolós (interfészes) modulja (pl. „Final Cartridge" vagy „Magic Formel") van, de sem ilyen csatolós rendszere (pl. Floppy-Speeder), sem hardver csatolója nincs, a „Starpainterrel" ugyancsak a User-Portot kénytelen kimenetként használni. Miután az installációs menüben beállította a használandó nyomtatót, tárolás után nyomja le a Reset-billentyűt. Utána töltse be a Starpaintert így:

LOAD "START",8,1
SYS 16384

Ezek után a modullal nyomtathat a User Porton keresztül.

Superquality

A „Superquality" programmal és Printfox nyomtatómeghajtóval egyes Printfox verzióknál problémák adódnak. Két POKE azonban csaknem mindig segít. Töltsük be a nyomtató meghajtót abszolút címre (8, 1-gyel), és adjuk be a két POKE utasítást:

POKE 25066,234
POKE 25067,234

Ezt követően tárolja újra lemezre a nyomtató meghajtót (pl. egy gépi kódú monitorral). A legtöbb nyomtatónál többé nem merül fel semmilyen probléma. Ha mégis, az — mint korábban — a Star LC—10-zel és a különféle hardver-csatolókkal fog előfordulni.

Giga-Publish CP80-nal

A Giga-Publish-nak a Melchers „CP80"-jához való illesztése céljából a programban a $7112 memóriahely értékét $13-ra kell változtatni (soremelés 19/216 inch). A nyomtatóillesztő programban az értékeket a következőképpen kell beállítani: 27,76 (grafika be); 27,51 (soremelés n/216); 27,64 (nyomtató reset); 13 (CR = kocsi vissza). Ezekkel a változtatásokkal a program kifogástalanul dolgozik. Figyelembe veendő azonban, hogy a laptervező (layout) menüben a jobb margó ne lépje túl a 2/3-os határt, mert a nyomtató csak 1280 pont/sor feloldóképességgel dolgozik.

Problémák a Star LC–10C-vel

A nagyon kedvelt Star LC—10C-vel, vagyis az LC—10-nek kifejezetten a C64-hez kifejlesztett változatával nagyon problémás esetek tudnak kialakulni az illesztésnél. Egyik az, hogy egyes szériáknál súlyos hiba van a nyomtató operációs rendszerében, ami miatt egy bizonyos grafikus parancsszekvencia esetén a nyomtatófej a jobboldali ütközőhöz vágódik. Másrészt nem realizálható (eltérően az előző NL—10 típustól), hogy a problémákat kiküszöböljék egyszerűen a Commodore-modulnak Centronics-verzióra való cseréjével.

Ha a leírt jelenség önnél is fellép, vagyis a nyomtatófej a jobboldali ütközőhöz csapódik, javasoljuk, hogy forduljon a kereskedőhöz.

FX–85 a 92008/G-vel

Még mindig nagyon elterjedt nyomtató az Epson FX-85. Ha ezt a Wiesemann-féle 92008/G csatolóval kötik a C64-hez, akkor sok grafikus programnál a nyomtatófej sok szükségtelen előre- és hátramozgást végez. Segítséget jelenthet itt a grafikus program betöltése előtt az alábbi parancssorozat beadása:

OPEN 1, 4, 1
PRINT #1
CLOSE 1
OPEN 1, 4, 3
PRINT #1
CLOSE 1

Ezek a parancsok a nyomtató kikapcsolásáig rögzítik a nyomtatási módot. Ezzel a fej rángatózása megszűnik, és ennek kellemes mellékhatásaként a grafika is hamarabb készül el.

Tisztelt Szerkesztőség!

A C64-esben az interpreter és az operációs rendszer ROM-jai, valamint az I/O RAM által fedve összesen 20 kilobyte olyan RAM található, amely a POKE utasítás számára csak részben, a PEEK függvény által pedig egyáltalán nem érhető el. Ehhez ugyanis ki kellene kapcsolni a ROM-okat, amellyel azonnal lehetetlenné tennénk a BASIC program további futását.

A mellékelt kis programmal a teljes 64 kB elérésére képes szubrutinokkal helyettesíthetjük az említett műveleteket, a betöltő elején feltüntetett módon. A PEEK, amelyben egész és valós típusú változóba is kérhetjük az eredményt, kb. 8%-kal lassabb az eredeti függvénynél, a POKE pedig kb. 15%-kal. A lassulást bizonyára a SYS címének kiértékelése okozza, ezt változóban érdemes tárolni. A program a $02A8—02FD területre töltődik. A betöltőben a program lemezre írásához a 'REM << ' jelzésig kell a sorokat törölni.

H. Gy.

```
; *** PEEK & POKE UNDER ROM *
; (C) DAVE - 9308
;
        .NOL
;
R6510     = $01
INTFLG    = $0E
LINNUM    = $14  ;/15
VARPNT    = $47  ;/48
FAC       = $61
;
FRMNUM    = $AD8A
CHKCOM    = $AEFD
GETSAD    = $B08B
GETPAR    = $B7EB
GETADR    = $B7F7
INTFLP    = $BC49
;
;
          * = $02A8
;
          JMP PEEK
;
; ------- POKE C,D
POKE      = *
          JSR CHKCOM
          JSR GETPAR      ; ,C,D
          TXA
          SEI
          INC R6510
          LDY #0
          STA (LINNUM),Y
          DEC R6510
          CLI
          RTS
;
;
; ------- PEEK C,V[%]
PEEK      = *
          JSR CHKCOM
          JSR FRMNUM
          JSR GETADR      ; ,C
          JSR CHKCOM
          JSR GETSAD      ; ,V
          SEI
          INC R6510
          LDY #0
          LDA (LINNUM),Y
          DEC R6510
          CLI
          LDX INTFLG
          BNE L1PE
;                   -- FLOAT (V)
          STY FAC+1
          STA FAC+2
          LDX #$90
          SEC
          JSR INTFLP
          LDA FAC+1       ;SIGN+
          AND #$7F
          STA FAC+1
          LDY #4
C0PE      LDA FAC,Y
          STA (VARPNT),Y
          DEY
          BPL C0PE
          RTS
;
;                   -- FIX (V%)
L1PE      TAX
          TYA
          STA (VARPNT),Y
          TXA
          INY
          STA (VARPNT),Y
          RTS
;
;
          .END
```

```
0 REM *** PEEK & POKE UNDER ROM *
1 REM (C) DAVE - 9308
2 REM * V=PEEK(C)  : SYS 680,C,V[%]
3 REM * POKE C,D   : SYS 683,C,D
4 REM << OPEN 2,8,2,"PEEK&POKE,P,W": REM 4,6,10,13
5 READ V,W: A=V+W*256
6 REM << PRINT#2,CHR$(V)CHR$(W);
7 FOR I=15 TO 20: PRINT I"]": READ V$,C
8 S=0: FOR J=1 TO LEN(V$) STEP 2
9 V=(ASC(MID$(V$,J))-65)*16+ASC(MID$(V$,J+1))-65
10 POKE A,V: A=A+1: REM << PRINT#2,CHR$(V);
11 S=S+V: NEXT J: IF S=C THEN NEXT I: GOTO 13
12 PRINT I"▒.SOR HIBAS"
13 REM << CLOSE 2
14 DATA 168,2
15 DATA EMLNACCAPNKOCAOLLHIKHIOGABKAAAJB,1970
16 DATA BEMGABFIGACAPNKOCAIKKNCAPHLHCAPN,1952
17 DATA KOCAILLAHIOGABKAAALBBEMGABFIKGAO,1696
18 DATA NABLIEGCIFGDKCJADICAEJLMKFGCCJHP,1783
19 DATA IFGCKAAELJGBAAJBEHIIBAPIGAKKJIJB,1856
20 DATA EHIKMIJBEHGA,721

READY.
```

Kedves Olvasóink!

Egy új kezdeményezésünket szeretnénk bemutatni és megmagyarázni. Többször is értesültünk olyan problémáról, hogy olvasóink — nyilván inkább a kezdők — számára komoly nehézséget okoz a BASIC programok szövegében levő vezérlőjelek azonosítása és megtalálása a billentyűzeten. Eltekintve attól, hogy semmiképp sem árthat ennek a húszhuszonöt különleges, jellegzetes alakú karakternek a megtanulása, mi mégis kedvében próbálunk járni tapasztalatlanabb olvasóinknak. A C64-esre elkészített programok egyikét-másikát ezentúl olyan alakban fogjuk megjelentetni, amelyben — egy program segítségével — a nehezen azonosítható karaktereket két-négybetűs rövidítésekkel helyettesítettük. Ennek persze feltétele, hogy a programot a szerző lemezen vagy kazettán is megküldje.

A színvezérlő jelek rövidítése a színkód sorrendjében a következő: Blk, Wht, Red, Cya, Pur, Grn, Blu, Yel, Org, Mar, Pnk, Gry1, Gry2, LGrn, LBlu, Gry3.

Kurzor- és képvezérlő jelek: Up, Dwn, Lft, Rgt, Clr, Hom, Ins, Del, RvOn, RvOf.

Egyéb vezérlőjelek és billentyűk, zárójelben a karakterkóddal: Stp (3, Stop), CCP (8), CCE (9), LF (10), Ret (13, Return), LCH (14), ESC (27), Spc (32, szóköz), NL (141), HCH (142), Pi (222 v. 255).

A rövidítések a szövegben szögletes zárójelben találhatók. Azokat a karaktereket, amelyek felismerése gondot okozhat, de nincs az előbbiekhez hasonló, értelemszerű elnevezése, a beírásukhoz szükséges billentyűk megjelölésével rövidítjük. Ilyen esetben az ,,s" a Shift, a ,,c" a Ctrl, az ,,=" a C-billentyűket jelölik. Például: [s—] a Shift+Mínusz billentyűpárost jelenti, hasonlóan értelmezhető a [cF] vagy a [=8] rövidítés is.

További könnyebbséget jelenthet a programok begépelésében, hogy az azonos karakterekből álló sorozatokat szintén rövidítve közöljük. Ilyenkor a szögletes zárójelben először a sorozat elemeinek száma látható, aztán egy kettőspontttal elválasztva következik az adott karakter. Például az ,,A[5:B]C" helyett a programba ezt kell beírni: ,,ABBBBBC".

Egy másik, eddig ismeretlen újdonság, hogy a hosszabb BASIC programokat egy önellenőrző kiegészítéssel együtt közöljük. Erre az késztet bennünket, hogy az olvasók, tudomásunk szerint, gyakran követnek el sajtóhibákat a program begépelése során, amelyeket aztán tapasztalat híján gyakran nem tudnak felfedezni, kijavítani. Ilyenkor előbb-utóbb a közölt program rejtett hibájára gyanakodnak, alaptalanul.

A hibaellenőrzés általunk támogatott módszere úgyanúgy az ellenőrző összeget használja, mint ahogy az a gépi kódú programok BASIC feltöltős alakjában már régóta bevett gyakorlat. Az ellenőrző rész mindig az eredeti program végére kerül, egy ,,— Checksum —" szövegű REM-sorral elválasztva. Ez a rész logikailag teljesen független a valódi programtól, így GOTO-val kell indítani. Indítás után ez a kis szakasz végigolvassa az őt megelőző programot, minden sorából képez egy számot, és összehasonlítja a DATA sorokban kódolva tárolt ellenőrző számmal. Ha minden sort hibátlanul írtunk be, persze a rövidítéseket is pontosan behelyettesítve, akkor ez a szám egyezik és nem kapunk hibajelzést, ellenkező esetben a képernyőn feltűnnek az eltérést mutató sorok sorszámai. Ha egy sorban egy betű helyett másikat írunk be, akkor 1/32 az esélye annak, hogy az ellenőrzésen ez a hiba átcsúszik. A javítás után újabb ellenőrzés kérhető. Amikor eljutottunk odáig, hogy nem kapunk több hibajelzést, az ellenőrző rész minden sora törölhető.

Azt a programot, amellyel a BASIC programok átalakítása, az ellenőrző összegek generálása és a program rendezett kinyomtatása végezhető, részletesen be fogjuk mutatni ,,Még többet ésszel!"

```
10 rem listazo peldaprogram
20 for i=1 to 8
30 : read a:read b
40 : print"[illegible]";chr$(a)
50 : print"[illegible]basic  listazo";chr$(b)
60 : print"[illegible]demo"
70 : for j=1 to 500:next j
80 next i
100 data 5,28,30,31,144,156,158,159,129
110 data 149,150,151,152,153,154,155

ready.
```

```
10 rem listazo peldaprogram
20 for i=1 to 8
30 : read a: read b
40 : print "[Clr][Hom][3:Dwn][3:Rgt]";chr$(a)
50 : print "[11:Rgt]basic  listazo";chr$(b)
60 : print "[Dwn][16:Rgt]demo"
70 : for j=1 to 500: next j
80 next i
100 data 5,28,30,31,144,156,158,159,129
110 data 149,150,151,152,153,154,155
111 rem ------ checksum ------
112 poke 65,peek(122): poke 66,peek(123)
113 v=peek(43)+peek(44)*256: read c$: p=0
114 n=peek(v+2)+peek(v+3)*256:
    if n=111 then end
115 p=p+1: if p>len(c$) then read c$: p=1
116 v=v+4: s=0: print n"[illegible]"
117 b=peek(v): v=v+1: if b then s=s+b: goto 117
118 if (s and 31)<>asc(mid$(c$,p))-64 then
    print "error in"n
119 goto 114
120 data "sie[s]9kw£"

ready.
```

című sorozatunkban. Azoknak a türelmetlen olvasóknak, akik addig is szeretnék kipróbálni a program szolgáltatásait, netán ilyen átalakított formátumban szeretnék programjaikat beküldeni, felhívjuk a figyelmét, hogy a program a Pötyögőszolgálattól **BASIC listázó** néven hamarosan megrendelhető.

a szerkesztő

Itt egy kis példával mutatjuk be a program működését.

Tegyük fel, hogy az 1. listán látható programot kívánjuk begépelni. Az újságban a 2. lista jelenik meg. Jól áttekinthető, hogy pl. a 60-as sorban egy ,,le'' után 16 ,,jobbra'' következik. Ha a 111—120 sorokat is bepötyögjük, GOTO 111 paranccsal ellenőrizhetünk.

Az ellenőrző sorok beírását is könnyíthetjük. Mivel ezeket minden programnál hasonlóak, írjuk be a 111—120 sorokat önmagukban és mentsük el. Legközelebb új formátumú program beírásakor először töltsük be az ellenőrzősorokat, javítsuk át őket az új listának megfelelően, majd gépejük be az új programot.

Info a Hollywood póker-hez

Az úgynevezett vetkőztetéses Pókerre sok számítógépes programot írtak. A HOLLYWOOD PÓKER főleg a minősége miatt érdemel kitüntetett figyelmet.

Sok változata van:

— a négy rész (ami a streap-tease girlök személyében és játékstílusában tér el) megtalálható egy két-lemezoldalas programban. Ezt forgalmazzák cartridge-ként is.
— de a négy részt feldolgozták önállóan (részenként kb. 100 blokk).
— sőt Isabell, Lorence, Denise és Stephanie néven csak a streap-tease képanyag is megtalálható 4 darab kb. 50 blokkos programban.

A HOLLYWOOD PÓKER népszerűségét mutatja, hogy a legtöbb számítógépen futtatható. Így

— C64, C+4, C16 és C116 (a két utóbbinál csak akkor, ha 64 kByte memória-kiegészítés is van).
— Atari ST
— Schneider CPC 464, CPC 664, CPC 6128
— Amiga

Póker szabályok

A pókernek nincsenek egységes szabályai. A HOLLYWOOD PÓKER az ötlapos, húzós pókert vitte gépre. Megtanulni eléggé könnyű. Az 52 lapos francia kártyával játszák. Ennek négy színe van: treff (fekete lóhere), pikk (fekete lándzsahegy), kőr (piros szív) és a káró (piros rombusz). Minden színben 13 különféle lap van. Ezek (erősorrendben): 2, 3, 4, 5, 6, 7, 8, 9, 10, bubi, dáma, király és az ász. A HOLLYWOOD PÓKER-nél két játékos van: a gép (valamelyik streap-tease girl) és az ember. A két játékos 100 DM induló pénzalappal kezd és ezt a gép ,,könyveli''. Ezt követően mindkét játékos 5—5 lapot kap. A cél: minél magasabb értékű figurát (kártyakombinációt) alakítani ki.

A figurák (erősorrendben):

Egy magányos lap, azaz nincs több olyan lap, amelyik erősebb figurát alkotna. A magányos lapok közül az az értékesebb, amelyik a lapok erősorrendjében magasabban áll.

— PAIR (pár). Két azonos lap. Például: 2 darab hármas, vagy 2 darab király. A magasabb lapokat tartalmazó pár rontja az alacsonyabbat. Két egyforma pár esetén az a jobb, amelyik mellett a három maradék lap között magasabb található.
— TWO PAIRS (Két pár). Két-két egyforma lapokból álló párok. Mindig a magasabb pár nyer. Ha a magasabb párok is egyenlők, akkor az ötödik lap dönt.
— THREE OF A KIND (hármas vagy drilling). Három egyforma lap. Például 3 darab hetes. A magasabb lapokat tartalmazó a jobb.
— STRAIGHT (ötös sorozat vagy kvint). 5 darab egymásközető, de bármilyen színű lap. Például: treff hetes. Káró nyolcas, kőr kilences, pikk tizes és káró bubi. Az győz, amelyik magasabb lappal végződik.
— FLUSH (kulőr). Öt egyszínű lap. Például: treff 2, 4, 6, 7, ász. A magasabb lapot tartalmazó nyer.
— FULL HOUSE (full hand vagy full). PAIR és TREE OF A KIND egyidőben. A két FULL HOUSE versenyét a TREE OF A KIND dönti el.
— FOUR OF A KIND (póker). Négy egyforma lap. Például 4 darab tizes. A magasabb lapot tartalmazó a jobb.
— STRAIGHT FLUSH (royal flöss). Öt egyszínű és egymást követő lap. Például: treff 4, 5, 6, 7, 8. A magasabb lappal végződő a jobb.

A játék során a magasabb figura legyőzi az alacsonyabbat. Azonos figuráknál a már ismertetett szabály érvényesül. Ha ez sem dönt, akkor a nyereményt felezik.

A játék célja: minél magasabb figurát hozni össze.

A kezdésnél kiosztott 5 lap alapján a játékosnak közvetkeztetnie kell a nyerési esélyeire és ennek alapján betehet a bankba egy 5—25 DM közötti összeget (BET). A bank neve: POT. Az ellenfélnek ugyanezt az összeget be kell tennie a bankba, de emelheti is a tétet (RAISE). Ezt a másik szintén beteszi, esetleg emeli.

Ha az egyik játékos nem akarja a tétet emelni (CALL), akkor a feleslegesnek tartott lapokat (amelyek nem illeszkednek valamely figurához vagy nem is egészülhetnek ki figurává) eldobhatja. Ezek helyett újabbakat kap és ezek esetleg jobb figurát eredményezhetnek. Például volt egy nyolcas PAIR. A másik három lapot cserélte. Kapott egy harmadik nyolcast, ami a nyerési esélyeit növelte.

Most ismét folytatódik a licitálás. Lehet a korábbi tétet növelni, amit a másik betesz és esetleg megnövel stb. Ha az egyik játékos már további növelésre nem hajlandó (CALL), akkor a lapokat összemérik és a bankot a nyertes kapja meg.

A licitálás közben bármikor és bármelyik játékos eldobhatja a lapjait (DROP). Például akkor, ha már újabb tétemelést nem akar kockáztatni. Ilyenkor, természetesen a másik játékos nyer.

A játék menete

A programot betölteni és indítani. A négy részt együttesen tartalmazó program esetében megválaszthatjuk az ellenfelet:

1. Izabelle
2. Lorence
3. Denise
4. Stephanie

A program kijelzi, hogy mennyi pénz van a bankban (POT) és mennyi pénze van a játékosnak (YOU), illetve a leánynak (ME)? Ezután a gép megkeveri a kártyát (I AM SHUFFLING). Mielőtt a lapokat kiosztaná 5-5 DM-et mindenkitől levon (ANTE). Majd megjelenik a játékos 5 lapja (YOUR CARDS).

Ezután — a már ismertett módon — a licitálásra kerül sor. Ha nem kívánjuk a tétet növelni, akkor CALL és a felesleges lapokat kicserélhetjük. A cserélendő lapok kiválasztása a joystick jobbra-balra mozgatásával történik. A kiválasztott lapok a tűz hatására átfordulnak. Ha minden felesleges lapot átfordítottunk, akkor a joysticket felfelé nyomva és a tűz gombot működtetve a csere megtörténik. A sorrend fontos: először fel és csak ezután a tűz. Megjegyzés: az egyes változatok vezérlése ettől eltérhet itt is és a későbbiekben is. Némi próbálkozással az eltéréseket kitapasztalhatjuk.

A lapcserét követően megtudjuk, hogy az ellenfél hány lapot cserélt (I TAKE … CARDS). Ez egy nagyon fontos információ (lásd. később!). Most az újabb licitálás következik. Megjegyzés: a licitálások során, ha nem akarjuk a tétet tovább növelni, választhatjuk a STAY-t is.

A második CALL után az ellenfél is bemutatja a kártyáit és a gép közli a játékos figurájának a nevét (YOU HAVE …), majd az ellenfélét (I HAVE …). Végül kijelzi a program, hogy ki nyerte a bankot: a játékos-e (YOU WIN POT), vagy az ellenfél-e (I WIN POT)?

Ha az ellenfél elvesztette az induló 100 DM-jét, akkor csak egy ruhadarabjának levetésével és zálogbaadásával kaphat újabb 100 DM-t. Később már hiába nyer, a levetett ruhadarabot nem kapja vissza.

A játék kétféleképpen fejeződik be: vagy a streap-tease girl vetkőzik le meztelenre, vagy a játékos veszti el a 100 DM-t.

Mindegyik leánynak 4 ruhadarabja van. Így a teljes streap-tease feltétele az, hogy a játékos 500 DM-t érjen el.

Mindegyik leánynak más-és-más a pókerezési taktikája! Tanácsos tehát ennek kiismerésére törekedni, mert enélkül az 500 DM összehozása reménytelen vállalkozás.

A póker művészete

A jó pókerezéshez két dolog kell.
— a lehetőségeink jó felmérése és
—az ellenfél tisztességes módon történő félrevezetése, azaz a blöffölés.

Ezek nélkül jó lapból is csak keveset hozhatunk ki, de ezekkel rossz lappal is nyerhetünk.

A lehetőségek jó felméréséhez bizonyos matematikai megfontolások szükségesek (vagy nagyon nagy gyakorlat). Számba kell venni, hogy valamilyen remélt figura a lapcserénél milyen valószínűséggel alakulhat ki. Ha például (színre való tekintet nélkül) a kezünkbe 8, 9, 10 és bubi van, az ötödik lap pedig nincs kapcsolatban ezekkel, akkor ez STRAIGHT-té alakulhat, de most semmit sem ér, mert az egyik vége hiányos. Mi a valószínűsége, hogy az ötödik lap cseréjével ez kiegészül? Így gondolkozhatunk:

— nekem 8 lap bármelyike jó lenne (akármelyik hetes vagy dáma)
— a kezembe levő 5 lapon kívül 47 lap van: 42 a talonban, amiből a felesleges ötödik lapom helyett újat kaphatok, 5 pedig az ellenfelemnél
— a „kinn levő" lapok 8/47-ed része, azaz 17,02%-a jó nekem
— de új lapot csak a talonból kaphatok. Feltételezhetem tehát, hogy a 42 talonlapból szintén 17,02% olyan ami jó a számomra.
— tehát 17,02% annak a valószínűsége, hogy egy lapot cserélve STRAGHT-em lesz. Ez eléggé biztató, tehát érdemes kockáztatni.

Gyakorlati szempontból ez a gondolkodási folyamat leegyszerűsíthető! Ha X-el jelölöm a számomra jó lapok számát, akkor

$$2{,}12765 \times X$$

azonnal megadja a valószínűség %-át. Az előbbi esetben 8×2,12765 = 17,02%. Vigyázat: ez a képlet csak arra az esetre vonatkozik, amikor egyetlen lapot cserélünk!

Egyetlen lap cseréjét feltételezve megadható egy táblázat is:

Kézben	Várakozás	Jó lap	Valószínűség (%)
Pair	Two pairs	6	12,77
Two pairs	Full house	4	8,51
Three of a kind	Four of a kind	1	2,13
Straight középen hiány	Straight	4	8,51
Straight végein hiány	Straight	8	12,02
Straight 1 végén hiány	Straight	4	8,51
Flush 1 hiány	Flush	9	19,15
4 magányos lap	Pair	12	25,53
Str.flush közepén hiány	Str. flush	1	2,13
Str.flush végein hiány	Str. flush	2	4,26
Str.flush 1 végén hiány	Str. flush	1	2,13

2 vagy 3 lap cseréjét tételezve fel a számítások bonyolultabbak. Két lapot cserélve:

Kézben	Várakozás	Valószínűség (%)
Pair	Two pairs	17,2
Pair	Three of a kind	7,8
Pair	Full house	0,8
Pair	Four of a kind	0,1
Pair	Bármilyen javulás	26,0
Three of a kind	Full house	6,1
Three of a kind	Four of a kind	4,3
Three of a kind	Bármilyen javulás	10,4
Flush (hiányos)	Flush	4,2

Három lapot cserélve:

Kézben	Várakozás	Valószínűség (%)
Pair	Two pairs	16,0
Pair	Three of a kind	11,4
Pair	Full house	1,0
Pair	Four of a kind	0,3
Pair	Bármilyen javulás	28,7
Flush (hiányos)	Flush	1,0

Azt sem árt tudni, hogy az első leosztásnál az egyes figurákra milyen valószínűséggel számíthatunk (%):

Magányos lapok	50,12
Pair	42,26
Two pairs	4,75
Three of a kind	2,11
Straight	0,39
Flush	0,20
Full house	0,14
Four of a kind	0,02
Straight flush	csaknem 0,08

A meggondolásoknak viszont nem csupán a saját lapokra, hanem az ellenfél lapjaira is ki kell terjedniök. Erre egyetlen támpont van: hány lapot cserél? Ha egyet sem, akkor esetleg mind az öt lapja rendezett (full house, flush, straight, straight flush). Két lap cseréje talán three of a kind-ra utal, ami viszont feljavulhat. Egy lap cseréje esetleges two Pairts-vagy hézagos, ötlapos figurát sejtet. Ha az ellenfél öt lapot kért, akkor minden lehet, mert semmi sem volt a kezében, de kérdés, hogy mi lett?

Itt jön be viszont a blöffölés, ami nélkül a póker nem póker.

A blöffölésről könyveket lehetne írni és valószínűleg írtak is. Itt csupán a lényeget lehet röviden összefoglalni:

— Ha már az osztás után jó lapunk van, akkor egy ideig ezt célszerű titkolni az óvatos licittel. Akkor kell a tétet jobban emelni, amikor már úgy tűnik az ellenfelünknek, hogy a lapunk nem valami ragyogó. Ilyenkor már az ellenfél fut a pénze után.

— Rossz lappal jó lapokat kell színlelni. A licitnél mindig emelni. Így talán az ellenfél a mienkénél jobb lapját is eldobja.

— A lapcsere is jó lehetőséget kínál a blöffölésre. Például egy király vagy ász párnál nem 3 lapot cserélünk, hanem csak kettőt vagy egyenesen egyet. Ezzel azt a látszatot kelthetjük, hogy a PAIR-nél lényegesen magasabb figurát kaptunk az osztásnál és azt még javítani akarjuk.

— A Pókert ,,élesben" nem ketten, hanem többen játszák. Ezeket a körülményeket a gép csak szimulálni tudja.

Végezetül egy jótanács: ha a HOLLYWOOD PÓKER-ben eredményesen játszunk a gép ellen, ez még nem ad alapot arra, hogy valódi pókerjátszmában is ,,villogtassuk" tudásunkat!

Marcsányi Zoltán

IBM PC

Softwareújdonságok

Kezdő IBM PC tulajdonosoknak bizonyára nagy problémát jelent a ,,bonyolult" DOS utasítások ,,betanulása", s ezek alkalmazása.

A filekezelő parancsok használata során előfordulhat, hogy az újdonsült PC felhasználók egy filekezelő parancs kiadása során megsemmisíthetik a drágán megszerzett programjaikat. Sokak talán ezért is félnek egy kicsit más gépről IBM-re áttérni.

A fájlok kezelésével (betöltés, másolás, átnevezés) kapcsolatos utasításokat egyszerűsíti meg számunkra Peter Norton programja, a NORTON COMMANDER. Ez idáig mindenki meg volt elégedve a régi 3.0-ás verzióval, de ez a tapasztalatunk 1993. május 10-én megváltozott, ugyanis elkészült a 4.0-ás Norton Commander, ami jócskán felülmúlta régi ,,testvérét". A kinézete szebb lett, mivel kétféle színösszeállításból választhatunk, de akinek monochrome monitora, vagy laptop gépe van, akkor ő az igények szerint is beállíthatja a képernyő színeit. De nemcsak kinézete, de tartalma is bőven változott.

A program figyelemreméltó előnyei a következők:

A legfontosabb talán az újabb s egyszerűbb fájlkezelés megemlítése, aminek segítségével egy becsomagolt (ARJ, ZIP stb.) fájlból kimásolhatunk, átnevezhetünk, vagy akár ki is törölhetünk fájlokat, mintha az adott fájl be sem lenne csomagolva.

Az F3 billentyű segítségével egy szövegfile, egy grafikát tartalmazó fájl, egy adatbázis vagy például egy Windows ikon (TXT, DOC, PCX, GIF, PIC, TIF, DBF, ICO) vagy egy becsomagolt fájl tartalma egy gombnyomásra lekérdezhető.

Új színeket használhatunk.

Lehetőségünk van bizonyos esetek (például a törlés) letiltására, s ezáltal programjaink túlélik az esetleges véletlen törlést.

Beállíthatjuk azt is, hogy a fent említett dolgokra a gép visszakérdezzen, tehát hogy: ,,Biztosan törölni akarod a ... fájlot?".

Ugyanígy például kikapcsolhatjuk a fájlok felülírását, s így még ha akarnánk is, akkor sem tudjuk felülírni a már létező fájlt.

Mint a régi változatban, itt is készíthetünk egy táblázatot olyan programokról, amiket sokszor használunk, s ezek már egy gombnyomásra futtathatók.

Most is beállíthatjuk, hogy egy ,,speciális kiterjesztésű fájlnál" mit csináljon a gép (például egy MOD fájlban tárolt zenét, — ha rámegyünk, és megnyomjuk az ENTER-t —, akkor azt lejátsza).

Ha netalán valahol elakadunk (bár ezt erőssen kételem), nyugodtan segítséget kérhetünk az F1 billentyű segítségével.

Az újdonságokból mára ennyi, s most lássuk a program használatát:

A könyvtárban az alkönyvtárak közül a kurzormozgató gombok segítségével közlekedhetünk, s az ENTER-rel választhatjuk ki azt. Ilyenkor meglátjuk, hogy az adott alkönyvtárban van-e újabb alkönyvtár vagy fájl, s azokat ugyanígy választhatjuk ki. A fájlkezelő DOS parancsok egy részét a funkcióbillentyűk használatával használhatjuk. Ezekkel részletesebben a későbbiekben foglalkozunk majd. Most azért röviden leírom a működésüket:

F1: Súgó, bármilyen szituációban segítséget ad.

F2: Ha akarjuk (lásd a következő részben), elkészíthetünk egy legördülő menüt, ami a legfontosabb programjaink címét tartalmazza (természetesen készíthetünk almenüt is: például játékok, játékok/logikai játékok, játékok/logikai játékok/Mahjong 4.), s ezek egy gombnyomásra betölthetők.

F3: Az egyes fájlok (szöveg, grafikai, táblázat stb.) tartalma nézhető meg vele.

F4: Az előbb említett fájlok tartalmát módosíthatjuk.

F5: Az INST vagy az egér jobb oldali gombjával kijelölt fájlok másolhatók vele.

F6: Ugyanezek a fájlok átnevezhetők, vagy áthelyezhetők.

F7: Könyvtárakat hozhatunk vele létre.

F8: Fájlokat és könyvtárakat törölhetünk vele.

F9: A későbbiekben fogunk róla beszélni.

F10: Kilépés a Norton Commanderből.

Hobbi **ELEKTRONIKA**

C-64 bővítések

HANGKAPCSOLÓ C64-GYEL

A C64 újabb érdekes felhasználására mutat példát ez a cikk, melynek ötlete a „A Commodore 64-es csatlakoztatási lehetőségei" című könyvből származik.

Emellett fontos és hasznos kiegészítője a korábban megjelent „A beszélő C64" című cikknek is. A két téma szoros kapcsolata adta az ötletet, hogy bizonyos konstrukciós megszorítás mellett ez a két áramkör egy panelre került.

A C-újság 93/7—8. számában részletesen kitértem arra, hogy ez miért történt. Az ismétlést elkerülendő, most csak annyit erről, hogy mindkét áramkör a joystick portra csatlakozik. A hangkapcsoló az 1. portra, a digitalizáló pedig a 2.-ra. Így az egyszerre feldugható két 9 pólusú canon csatlakozó stabilan tartja a lábai közé beforrasztott panelt, nem kell külön konstrukciós problémával foglalkozni.

A két független áramkör ezért került egymással rokonságba, egy műanyag cartridgedobozba zárva. Természetesen mindkettőnek külön dokumentációja, ültetési, kapcsolási rajza és árajánlata van. Lásd a cikk utáni hirdetést.

A hangkapcsoló felhasználása igen sokrétű lehet, könnyen alkalmazható egyedi igényekre is. A működésnek az a lényege, hogy valamilyen jelforrást kötünk a bemenetére és a számítógép figyeli a kimenetet. A bemenőjel lehet egy jelgenerátor, mikrofon, egy passzív infra érzékelő stb. Ha a bemenetre valamilyen jel kerül, billen az áramkör. Ezt a számítógép érzékeli, feldolgozza és a szoftver szerint valamilyen parancsot adhat.

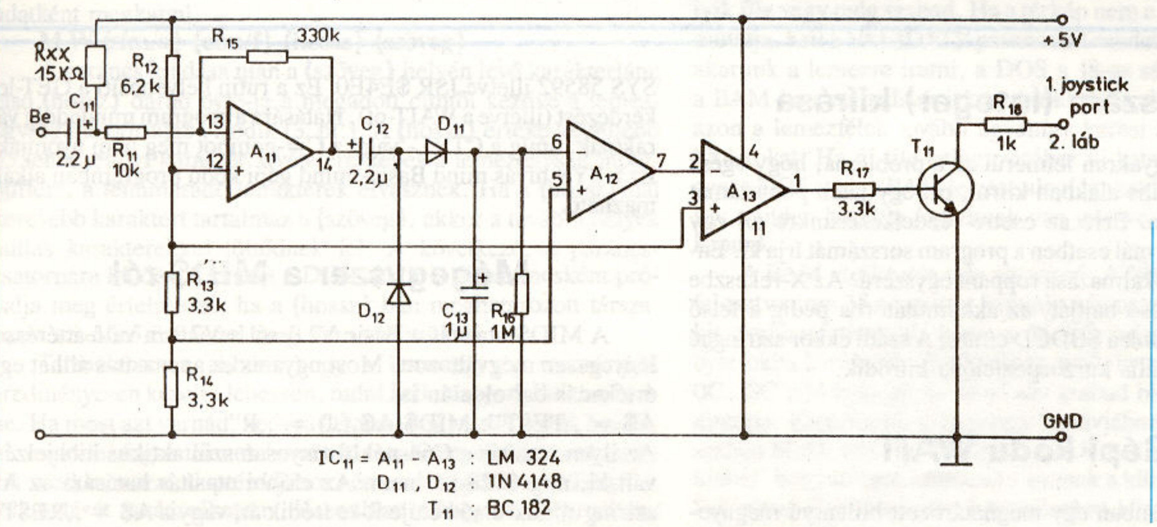
Ezt az alaphelyzetet finomítani lehet úgy, hogy az áramkör csak egy bizonyos frekvenciára, kódra legyen érzékeny: például tapsra, üvegcsörömpölésre stb. Természetesen ilyenkor ennek megfelelően változik az érzékelőköre, hiszen szelektívvé kell tenni az áramkört.

Ezzel a lehetőséggel most nem foglalkozunk, csak az alapesetet vizsgáljuk részletesen.

Az áramkör működése

Az 1. ábrán látható kapcsolásban a bemenőjel az R11, C11 tagon át jut az A11 műveleti erősítőre. Ez az alacsony szintű jelet felerősíti. Az erősítés mértékét az R15/R11 hányados határozza meg. Az erősítő tápfeszültsége +5 V aszimmetrikus, ezért a munkaponti beállítást az R12, R13 és R14 végzi. Az R12 és R13 közös pontján közel fél tápfeszültség mérhető. Nyugalmi helyzetben ez jelenik meg az A11 kimenetén. Ez meghatározza a kivezérlési tartományt is, bár a túlvezérlésnek itt nincs jelentősége.

A jel az erősítőről egy feszültségkétszerező egyenirányítóra kerül. Ennek C12, D12 tagja tulajdonképpen szinteltolóként működik. Az utána kapcsolt D11 az eltolt szintű jelet egyenirányítja, C13-at feltölti. Ha a C13 feszültsége eléri, vagy meghaladja az R13, R14 közös pontján mérhető feszültséget, az A1 komparátor átbillen és mindaddig úgy marad, amíg a C13 feszültsége a komparálási szint alá csökken. Ebben segít az R16-os ellenállás, ami kisüti C13-at. Az A13 invertáló fokozat. Azért van rá szükség, hogy a T11 alaphelyzetben zárva legyen, a komparátor billenése után pedig kinyisson. A tranzisztor kollektora az R16-on át az 1. joystick port 2. lábára csatlakozik. A gép ennek a változását figyeli.



1. ábra
A hangkapcsoló kapcsolási rajza

A hangkapcsoló szerelése, élesztése

A szerelést a nyomtatott áramkör elkészítésével kezdje. Ennek a rajza a 93/7—8. sz. újságban jelent meg. A panel ez alapján pozitív 20-as technológiával elkészíthető. Ha ez gondot jelent, jöjjön el az üzletembe, ahol a mintadarab megtekintésével egybekötve megvásárolhatja az áramkör készre szerelt változatát, egységcsomagját, vagy részegységeit, ahogy Önnek kedvezőbb. Cím a cikk utáni hirdetésben megtalálható.

Már említettem, hogy ez az áramkör a hangdigitalizálóval közös panelen kapott helyet. Ezért, hogy az alkatrészeket pozíciószámuk alapján ne lehessen összekeverni, ennél az áramkörnél a számozás tízzel kezdődik. Ha valaki már megépítette a hangdigitalizálót és most ezzel bővíteni szeretné, nem sok teendője van. Az üzletemben vásárolt egységcsomagból a 2. ábra alapján ültesse be az alkatrészeket. Valamilyen kézműszerrel mérje meg a műveleti erősítők munkapontjait. Ellenőrizze, hogy a féltápfesz a szövegben jelzett pontokon megvan-e. Kössön valamilyen jelforrást a bemenetre és ellenőrizze az A13 kimenetének szintjét. A bemenőjel szintjétől függően itt 5V-ot, vagy 0V-ot mérhet. Ha ez teljesül, működőképes az áramkör. Olvassa be lemezről vagy kazettáról a szoftvert, amit az egységcsomagban talált. Ez a kísérletezéshez tökéletesen megfelel és ötletszinten más, célorientált feladatra felhasználható.

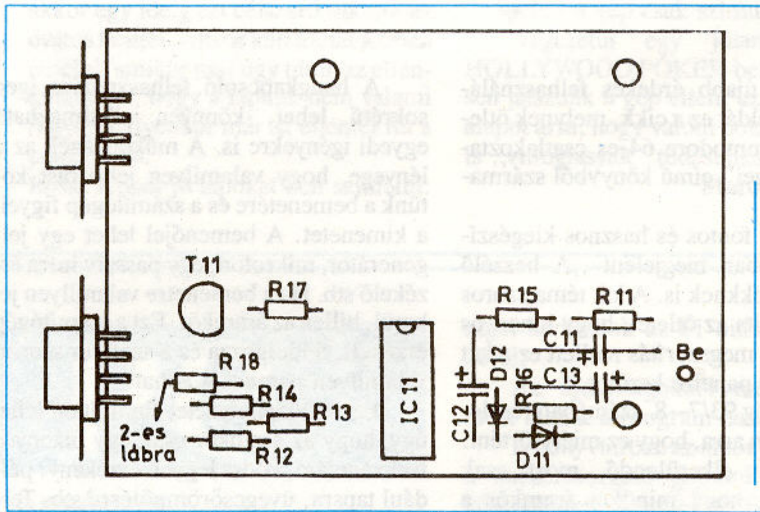
Ha kondenzátormikrofont használ, a bemenetre előfeszítő ellenállást kell kötni. Ennek nincs helye a panelen. A kapcsolási rajzon Rxx jelöli a bemenet és a +5 V-os tápfesz között. Értéke 15 KΩ.

Ugyanez a megoldás használható a hangdigitalizálónál is. Érdekes felhasználási lehetőséget nyújt a digitalizáló és a hangkapcsoló együttes kombinációja az őrzésvédelemben. A hangkapcsoló által érzékelt zajra vagy más érzékelő jelére a C64 valamilyen digitalizált szöveggel válaszolhat.

Ha Ön nem ismeri a Hangdigitalizáló cikket, lapozza fel a C-újság 93/7—8. számát, ott részletes szerelési útmutatót talál, ami mindkét áramkörre érvényes.

Folyamatosan kapható egységcsomag, panel és részegység, a C-újságban korábban megjelent cikkeimhez is. Ezek egy része működés közben megtekinthető, kipróbálható.

A vidéki olvasóknak segít a szerző levelező-egységcsomagküldő szolgáltatása. A megrendelt csomagot postán, utánvéttel elküldöm. Telefonon és levélben is rendelhet. A HOBBI ELEKTRONIKÁHOZ nem kell hosszú levél. rendelését röviden, egyértelműen közölje.



2. ábra
A hangkapcsoló ültetési rajza

A hangkapcsoló árajánlata

A Hangkapcsoló működő mintadarabja megtekinthető, kipróbálható, részegységként, egységcsomagként vagy készre szerelve megvásárolható a szerző üzletében.

Ára egységcsomagban 750,— Ft, a hangdigitalizálóval együtt 1350,— Ft.

Készre szerelve 990,— Ft, a hangdigitalizálóval együtt 1900,— Ft.

Külön vásárolható elemek: panel 200,— Ft, doboz 150,— Ft.

Azt üzletben beszerezhetők a C-újság korábbi számai is.

Vásárláskor használja a HOBBI ELEKTRONIKA kuponját.

Levélcím: HOBBI ELEKTRONIKA
1656 BUDAPEST Pf. 50.

Üzletcím: BUDAPEST VII.
Dózsa György út 16.
Jobbágy u. sarok.

Nyitva: H—P. 10—17-ig,
zárás után üzenetrögzítő.
Tel./Fax: 122-8892

Egész szám (integer) kiírása

Gépi kódban gyakran felmerül az a probléma, hogy egész számot kell decimális alakban kiírni, pl. egy játék pontszámát kiírni a képernyőre. Erre az esetre rendelkezésünkre áll egy ROM-rutin ami normál esetben a program sorszámát írja ki. Ennek a rutinnak az alkalmazása roppant egyszerű. Az X-rekeszbe betöltjük a szám alsó bájtját, az akkumulátorba pedig a felső bájtját és ezután ugrunk a $BDCD címre. A szám ekkor sztringgé változik és az aktuális kurzorpozícióba kiíródik.

Gépi kódú WAIT

Ha egy programban egy meghatározott billentyű megnyomására kell várakozni, jól alkalmazható a következő ROM-rutin: SYS 58592 illetve JSR $E4E0. Ez a rutin helyettesíti a GET-lekérdezést (illetve a WAIT-ot). Hatására a program mindaddig várakozik, amíg a CTRL- vagy a C=-gombot meg nem nyomják. Ez a SYS-hívás mind Basic- mind gépi kódú programban alkalmazható.

Mégegyszer a MID$-ról

A MID$ utasítás a Basic V2.0-ról a V7.0-ra való áttéréssel lényegesen megváltozott. Most ugyanis ez az utasítás állhat egy értékadás bal oldalán is.

A$ = „TEST": MID$(A$,1,1) = „R"

Az ilyen utasítás a C64-nél bizonyosan szintaktikus hibajelzést vált ki, míg C128-nál nem. Az előbbi utasítás hatására az A$ sztringváltozó első betűje R-re íródik át, vagyis A$ = „REST" lesz.

Még többet ésszel!

XIX. rész

A lemezegységben egy közel 16 kilobyte-os ROM-ban van a DOS, amelyet további 2 kB RAM is kiszolgál. Ezek elérésére, hasonlóan a BASIC nyelv PEEK és POKE műveleteihez, létezik két DOS-parancs: a Memory-Read és a Memory-Write.

Több különbség van ezek és a direkt állományok kezelésére használt parancsok között. Elsősorban elmarad a külön puffer megnyitása, az adatok a parancs-csatornán közlekednek. Más a paraméterek típusa is, ugyanis itt sosem számokat, hanem meghatározott kódú karaktereket kell megadni. A parancsoknak csak a rövidített alakja használható, és kettőspont nem írható utánuk. A leírásban levő szóközök is csak az olvashatóságot szolgálják, kiírni őket nem szabad, hiszen a DOS nem tudja, hogy a szóközzel nem egy 32-es (ez a szóköz kódja) értékű paramétert adtunk át.

A ⟨cím/a⟩ és ⟨cím/f⟩ nevű paraméterek olyan karaktereket jelentenek, amelyek kódjai alsó/felső byte sorrendben mutatnak az írás vagy olvasás során elsőként érintendő memóriacímre. Ha a memóriacím jele MC, a két összetevő jele CA és CF, akkor egymásba az 1. példán látható módon számíthatók át.

M-R ⟨cím/a⟩ ⟨cím/f⟩ [⟨hossz⟩]

A megadott memóriacímtől kezdve ⟨hossz⟩ számú egymás utáni byte olvasható ki a parancs-csatornán keresztül. A ⟨hossz⟩ értéke legfeljebb 255 lehet. Az adatok után egy plusz Return karaktert (13-as kód) is kapunk, azért, hogy INPUT # utasítással is átvehető legyen az adatsor, egyetlen karakterláncban. A „File Vége" jelzés a Return kiolvasásakor jelenik meg. Az INPUT # használatának feltétele, hogy a szöveg 88 karakternél ne legyen hosszabb, és Return, vessző, kettőspont vagy nullás karakter zárja le. Hátránya, hogy ha ezek bármelyike az üzenet közben is előfordult, akkor a további adatok elvesznek. Bár lassabb, de sokszor biztosabb a 2. példa mintájára GET # utasítást használni. A példa egyébként összevethető az előző rész 3. példájával is.

Ha a ⟨hossz⟩ karakterét elhagyjuk (a szögletes zárójel most is e választási lehetőség jele, és nem a parancs része), akkor a DOS az értékét 1-nek feltételezi. Ha a megadott hosszon túl olvasunk, akkor a DOS szokásos hibaüzenetének karaktereit fogjuk adatként megkapni.

M-W ⟨cím/a⟩ ⟨cím/f⟩ ⟨hossz⟩ ⟨szöveg⟩

A parancs kiadása után a ⟨szöveg⟩ helyén levő karakterlánc első ⟨hossz⟩ darab byte-ja a megadott címtől kezdve a lemezegység memóriájába íródik (3. pl.). A ⟨hossz⟩ értéke legfeljebb 35 lehet, ennyi karakter átvételére képes a lemezegység inputpuffere, a fennmaradó karakterek elvesznek. Ha a ⟨hossz⟩-nál kevesebb karaktert tartalmaz a ⟨szöveg⟩, akkor a további helyek nullás karakterekkel töltődnek fel. A következő, a parancscsatornára küldött üzenetet a DOS akkor is új parancsként próbálja meg értelmezni, ha a ⟨hossz⟩-ban meghatározott társzakaszt az előző üzenettel még nem töltöttük ki.

Természetesen ahhoz, hogy a lemezegység memóriáját eredményesen kezelni lehessen, tudni kell, hogy mi hol van benne. Ha most azt várnád, kedves Olvasó, hogy erről fogok beszélni, csalódni fogsz. Ennek bemutatása már egyáltalán nem tartozik a cikksorozat alaptémájához, és csak régebbi szakkönyvek felkutatását tanácsolhatom. Hasonlóan elhagyom még néhány olyan DOS-parancs ismertetését is, amely szintén elvezetne bennünket a BASIC programozástól. Egy parancs van még hátra, azt majd a relatív file kapcsán fogom megemlíteni.

Beszélni fogok viszont a lemez felosztásáról, ahogy megígértem. A lemez általában teljesen egyenrangú sávokból és szektorokból áll. Kivétel a 18-as sáv, amelyet a DOS, ha másra nem kényszerítjük, teljes egészében kiemelt fontosságú adatok számára tart fenn. Ilyen adat például a BAM, a lemez fejléce és a katalógus.

A 4. példa az egyik lemezem 18-as sávjának 0. szektorát ábrázolja, nagyjából úgy, ahogy azt a korábban ismertetett lemezmonitorral láthatom. Kaotikusnak tűnhet, pedig nagy itt a rend. Minden adat mindig csak a saját, szigorúan meghatározott helyére kerülhet. A byte-ok, amint abban a direkt állományok puffereinél (ez is csak az!) megegyeztünk, 0 és 255 közötti sorszámot kapnak. A monitor képén tizenhatos számrendszerben minden sor elején a sor első adatbyte-jának sorszáma látszik, egy sor 8 byte-ot tartalmaz. Jobbra van az adatok képszerű megjelenítése. Sorban elmondom minden byte-ról, hogy mi a szerepe.

00—01. byte: a katalógus első blokkjának helyét mutatja a lemezen, itt tehát 18. sáv 1. szektor. Az a helyzet, hogy bármi más értéket írnánk is ide, a DOS mindenképpen a 18—01-es blokknál keresi a katalógus elejét, tehát ennek nincs a világon semmi jelentősége. Az első validálás vissza is állítja a helyes értéket, bár előtte az itt jelzett „katalógusblokkot" lefoglalja a BAM-ban.

02. byte: a lemez formátumtípusának a jelzése. Ez még abból az időből maradhatott fenn, amikor ezt a lemezformátumot más gépek lemezegységei is használták, így a lemez beosztásának azonosítása még lényeges lehetett. Ma már nincs ilyen szerepe. Ennek ellenére nehogy bárki is megpróbáljon ide mást írni! Ha ugyanis akár csak véletlenül is $41-től eltérő érték kerül ide, a lemezre soha többé nem tudunk semmit írni! Csak a formázás tudja ezt az írásvédelmet felülbírálni, meg szerencsére egy-két nagyon ravasz lemezkezelő program, mint például az Underground Network nevű (?) monitor. Eltérő azonosító érték esetén az első író műveletre 73-as hibaüzenetet kapunk.

03. byte: értéke 00, nincs szerepe.

04—8F. byte: a BAM (Block Availability Map), magyarul blokkfoglaltsági térképnek hívják. Ez az a hely, ahol a DOS minden blokkról nyilvántartja, hogy felhasználta-e már valamelyik file vagy még szabad. Ha a térkép nem a valóságos helyzetet mutatja, kell a VALIDATE paranccsal rendet rakni. Ha egy file-t akarunk a lemezre iratni, a DOS a 18-as sávhoz legközelebbi, a BAM-ban szabadnak jelzett blokkba ír először, majd ugyanazon a lemezfélen tovább ugyanígy keresi a következő szabad blokkokat. Ha új file bejegyzéséhez új katalógusblokk szükséges, a DOS azt is a BAM jelzése alapján veszi fel. Ha a katalógussáv minden blokkja foglaltnak van jelezve, 72-es hibaüzenetet kapunk.

A BAM szerkezete elég egyszerű. A 140 byte-ot négyesével felosztva pont 35 csoportot kapunk, a lemezen is 35 sáv van. Így kiszámítható például, hogy a 18. sáv adatai a $48—4B jelű byte-okba kerülnek. Minden csoport 0. byte-ja (a blokk 04, 08, 0C...8C jelű byte-ja) az illető sáv szabad blokkjainak a számát mutatja. Eszerint pl. a lemezem 5. sávjában már csak 19 ($13) szabad blokk van. Így végignézve a lemez kihasználtságát, feltűnhet, hogy milyen hézagosan vannak a külső sávok elfoglalva. Sok rövid életű állomány használatával könnyen jön létre ilyen helyzet.

```
(1.)
1 MC=CA+CF*256
2 CF=INT(MC/256): CA=MC-CF*256
```

```
(2.)
0 REM *** LEMEZNEV *
2 OPEN 15,8,15,"I0"
3 PRINT#15,"M-R";CHR$(144);CHR$(7);CHR$(16)
4 N$="": FOR I=1 TO 16: GET #15,V$
5 N$=N$+V$: NEXT I: PRINT "A LEMEZ NEVE: ";N$
9 CLOSE 15
```

```
(3.)
0 REM *** EGYSEGSZAM D0->D1 *
1 D0=8: D1=9
2 OPEN 15,D0,15,"M-W";CHR$(119);CHR$(0);CHR$(2);CHR$(D1+32)+CHR$(D1+64)
3 CLOSE 15
```

```
(4.)
>R 12 00
>:00  12 01 41 00 14 FF FB 1F
>:08  11 5F FF 15 10 F5 D5 1F
>:10  15 FF FF 1F 13 FD F7 1F
>:18  12 BF CF 1F 13 F7 DF 1F
>:20  10 FF 3D 15 12 AF FF 1E
>:28  10 7F 7D 15 11 FE 7E 1B
>:30  12 FF FE 0B 0E 56 7B 1D
>:38  00 00 00 00 00 00 00 00
>:40  00 00 00 00 00 00 00 00
>:48  12 FE FF 07 00 00 00 00
>:50  00 00 00 00 00 00 00 00
>:58  00 00 00 00 00 00 00 00
>:60  0F FE FC 03 09 38 F3 00
>:68  10 FD FD 03 0D EC 6F 03
>:70  12 FF FF 03 12 FF FF 03
>:78  12 FF FF 03 11 FF FF 01
>:80  11 FF FF 01 10 FF FD 01
>:88  11 FF FF 01 11 FF FF 01
>:90  44 41 56 45 20 2F 41 A0
>:98  A0 A0 A0 A0 A0 A0 A0 A0
>:A0  A0 A0 54 30 A0 32 41 A0
>:A8  A0 A0 A0 00 00 00 00 00
 ...
>:F8  00 00 00 00 00 00 00 00
```

```
      (5.)
   14 ($0E) SZABAD BLOKK

      56        7B        1D
   01010110  01111011  00011101

              1111      211  1
   6 4 21    4321 98    098 6
```

A négyesek maradék byte-jai mutatják a sáv minden blokkjáról, hogy foglalt-e vagy sem. Minden blokknak egy-egy bit felel meg, a szabad állapotot a bit 1-es értéke jelenti. A négyesek 1. számú byte-ja (a blokkban 05., 09. stb.) a 7—0. blokk foglaltságát jelzi (a helyiérték szerint), a 2. byte-ok a 15—8. blokkokét, a 3. byte-ok a 23—16. blokkokét. Mivel egy sávon sincs annyi blokk, amennyi adata a három byte-ban elférne, a nem létező blokkok természetesen foglaltnak vannak jelölve. Az 5. példán a blokk 34—37. byte-jait kiemelve a 13. sáv foglaltsági adatait részletezem. Ha valamelyik csoportban nem pont annyi 1-es bit van, amennyit a csoport 0. byte-ja mutat, akkor a legközelebbi íráskor 71-es hibaüzenetet kapunk.

90—9F. byte: a lemez neve. Mindig 16 byte hosszú, a hiányzó karakterek helyére 160-as karakterek (shift-szóköz) kerülnek.

A0—A1. byte: értékű $A0, szerepük nincs.

A2—A3. byte: az azonosító karakterei. A formázáskor ezeket kell a név mellett megadni.

A4—A6. byte: formázáskor az itt látható „2A" szöveggel töltődik fel. Régebben szintén a lemezformátum azonosításában lehetett valami szerepe, ma már nincs jelentése. Mivel a tartalomjegyzék kiírásakor megjelenik, lehetőségünk van a lemez azonosítóját összesen öt karakter hosszúra megnövelni, monitorral vagy más közvetlen módon a lemezre írva.

A7—AA. byte: értékű $A0, szerepük nincs.

AB—FF. byte: értékük $00, szerepük nincs. Ez utóbbi két területet semmilyen DOS művelet nem érinti (a formázás kivételével), ezért kiegészítő adatok, megjegyzések tárolására alkalmasak.

Legközelebb a katalógus elemzésével folytatjuk.

Hódi Gyula

POKE-ok a grafikához

Amikor a C128-nál saját karaktereket definiálunk, gyakran problémát okoz, hogy hogyan kapcsoljuk azt be. Ha ezek a karakterek a memóriában a $3000 (decimális 12288) memóriahelytől kezdve vannak elhelyezve, akkor erre a karakterkészletre POKE 2604, (PEEK (2604) AND 240) OR 12 utasítással kapcsolhatunk át.

Egyszerű trükkel gondoskodhatunk arról, hogy a számítógép gyorsabban fusson. Ehhez egyszerűen 60 Hz-re kapcsol át. Az utasítás:
POKE 2563,0
A lassabb működés (50 Hz)
POKE 563,255
utasítással érhető el. Ez a trükk különösen jól felismerhető, ha a 80 karakteres képernyő minden memóriahelyét inicializáljuk a SYS 57721: REM JSR $E179
utasítással.

TCT program

Konfiguráció: C-64/MPS-1230

I. Bevezető

A címbéli „mozaikszó” feloldása: Three-Cleft Text. (Háromhasábos szöveg.) Hogy e kifejezés mit ígér a gyakorlatban, az a jelen printeroutputból máris sejthető.*

Mielőtt rátérnék a program bemutatására — ismeretlenül bár, de — köszönetet kell mondanom Horváth László (alias „HORLA-SOFT”) úrnak a Commodore újságban megjelent cikksorozatáért.

Ugyanis a TCT program megírásának előzménye az volt, hogy jónéhány hónapja olyan levelet kaptam külföldről, amelyet két hasábot nyomtató szövegszerkesztővel írtak. „Ide nekem az oroszlánt is!” kiáltottam fel. Azután ebben is maradt a dolog. A printer programozásával még csak-csak megbirkóztam volna, de a szövegszerkesztő rész...? Így hát „hibernáltam” a jámbor óhaj szintjén maradt ötletet.

A reanimáció akkor következett be, amikor a C-újság 1991/6. számában elolvastam Horváth úr cikkét, amely többek között beígérte egy „mini iratkészítő” közlését.

Már az assembly lista tanulmányozása is adott hasznos ötletet! Nevezetesen a CTRL-kódtábla módosításának megoldását — némi racionalizálással — felhasználtam.

Az igazi „kulcsot” azonban a 7/8. számban közölt folytatás adta! A sorok tömbelemként való kezelése — legalábbis számomra — a reveláció erejével hatott. Nem tudván megvárni a szeptemberi számot, a BASIC betöltővel létrehoztam a LINEIN és a DIR gépi kódú rutinokat, majd ezután ezek nyomtatott assembly listáját. (A módszer leírása megjelent a C-újság 1992/3. számának 29. oldalán.)

Nekilátva az érdemi munkának — az alapcélon túl — három szempontot tartottam fontosnak:

1. Gyors programfutás.
2. Kényelmes használat.
3. Biztonság. (Védelem a „helyrehozhatatlan” tévedések ellen.)

*A kézirat háromhasábos nyomtatással készült.

```
10 rem main prg./tct 920317
20 :
30 if a then 80
40 poke 53280,6:poke 53281,8:poke 646,9:poke 808,234
50 print"█chr$(14)chr$(8)
60 print"██████████     █ T H R E E - C L E F T   T E X T "
70 a=1:load"gepi*",8,1
80 sys 51200
90 dim a$(1),t$(354),h(3),hb(26),k(9),sh(3),bt$(3):r=354
100 h(1)=37:h(2)=17:h(3)=11:sh(1)=64:sh(2)=40:sh(3)=40
110 bt$(1)="PICA":bt$(2)="ELIT":bt$(3)="CONDENSED"
120 hv$="█:for l=1 to 40:hv$=hv$+"-":next l:hv$=hv$+"█
130 for l=1 to 40:pu$=pu$+" ":next l:ec$=chr$(27)
140 lm(1)=8:lm(2)=7:lm(3)=5:p$=chr$(160):u$="█>":goto 880
150 fu=0:if s+(sk*bt)<r then 190
160 print "██       █ A mem_ri rban nincs t_bb hely ! █
170 print "██       <F-1>":fu=1:gosub 530
180 wait 198,1:get w$:if w$<>chr$(133) then 180
190 return
200 for a=1 to bt:q$=t$(w):if uz=4 then 220
210 h=h(bt):if q$=p$ then q$="█+str$(w)+" █:h=h+2
220 pr$=pr$+left$(q$+pu$,h)+d$
230 w=w+m:next a:w=w-m:if uz=4 then 250
240 print left$(pr$,39):goto 260
250 print#1,pr$
260 return
270 i=0:sx=sk:z=s
280 if z/bt<>int(z/bt) then z=z+1:t$(z)=p$:goto 280
290 m=59-sx:sm=m*bt:if (z-i)=>sm then 310
300 m=(z-i)/bt
310 hv=m+i:p=0:sx=0:return
320 q$=t$(i):if q$<>p$ then 340
330 q$="█+str$(i)+"":if i=hi then q$=" * "+q$:goto 350
340 if i=hi then f=peek(214):print"█;
350 print q$"█;
360 if len(q$)<40 or bt=1 then print
370 if uz=2 then 390
380 if i=0 and sk then for j=1 to sk+1:print u$:next j
390 return
400 for dl=f to 24:poke 781,dl:sys 59903:next dl
410 poke 214,f:poke 211,v:sys 58640:return
420 print"█     █ Eg-szen biztos benne? █:goto 440
430 print"     O.K. ? ? ?";
440 print" █I█gen/█N█em":gosub 530
450 get w$:if w$<>"i" and w$<>"n" then 450
460 return
470 gosub 440:gosub 410
480 f=f+2:if w$="i" then print " *";
490 print "  "
500 return
510 open15,8,15:input#15,e,e$:if e=0 then return
520 print"█     █ "e$" █:close2:print#15,"i":close15:return
530 sys 52011:return
540 print"████     █ K-rem ellen_r_zze:██
550 print"     A nyomtat_ _zem rllapot rt.█:f=16
560 print"     A pap_r mennyis-g-t."
570 print"█     A lap pozicion rl rs rt!██
580 gosub 430:if w$="n" then hb=1:return
```

II. A program szerkezete és felépítése

Elsőként a DIR rutint egészítettem ki. Így hosszabb lemezkatalógus esetén sem ,,szalad el" a tartalomjegyzék eleje, hanem a képernyő beteltével megáll a kiírás és csak ⟨RETURN⟩-re folytatódik.

A LINEIN rutint viszont ,,elfelejtettem". Három oka volt ennek:

1. A lemezolvasás sebességét a soros port eleve limitálja, tehát a gépi kódú rutin nem tud gyorsítani.
2. Az írott szövegben előforduló — a DOS által elválasztójelként (:,;) értelmezett — karakterek megjelenítése és reprodukálhatósága más módon is megoldható.
3. Végül, de nem utolsósorban a ,,csonka sorok", tehát a max. sorhossznál rövidebb sorok nem rabolnak el felesleges lemezkapacitást. (Ennek a leírásnak az első 2 oldala 45 lemezblokkot igényel.)

A program futásának gyorsítása érdekében az operációs rendszer Garbage Collection rutinja is talonba került. Ugyanis a stringműveletenkénti meghívás ellenére, átlagban 250 soronként automatikusan végrehajtódott egy ,,szemétgyűjtés". Nem volt értelme tehát, hogy a programból hívogassuk a rutint.

A program tesztelése, ezen leírás elkészítésével történt. A felfedett hibák, vagy az esetleges újabb ötletek nemcsak a program, hanem a leírás módosítását is szükségessé tették.

Eközben ,,derült ki", hogy a beszúrási és sortörlési művelet BASIC-ben megengedhetetlenül lassú. (A leírás 2. sora elé történő beszúrás kb. 30 sec.-ot vett igénybe.) Így ezt a feladatot is gépi kódú rutinra kellett bízni. A végrehajtási idő 0,5—1,5 sec.-ra csökkent.

Ugyancsak a programfutás gyorsításának igénye miatt, egy gépi kódú rutint kellet kreálni, amely a beírt karakterek bevételezésén túl, a ,,tiltott" karakterek konverzióját is végrehajtja (GET).

Az elfogadott karaktereket a rutin elhelyezi a saját pufferében, majd a sor befejeztével, ennek címét adja meg az A$(0) stringváltozó címeként. Visszatérve a BASIC-be: T$(X) = A$(0). A keresési procedúrát elkerülendő, elsőként az A$(1) ,,tömböt", majd a T$(354)-es tömböt kell deklarálni. Így azok címei relatíve fix helyen találhatók.

A teljes — 7 blokkos — gépi kódú program valójában több rutinból áll. Ezek sorrendben:

```
590 if peek(52222) then 660
600 print"█ █ K┼rem ┌ll▬tsa be a SET UP elj┌r┌ssal:█
610 print"     1./az EPSON FX-80 emul┌ci▬t,█
620 print"     2./a 4-es egys┼gsz┌mot  ├PC├re.█
630 print"     3./  ├ENABLE D.L.L.; YES █:gosub 430
640 if w$="n" then hb=1:return
650 sys 52166:poke 52222,1:close1
660 close1:open1,4:print#1,chr$(17)chr$(24);:return
670 if ci then 720
680 print#1,ec$"P";chr$(18);
690 print#1,ec$"E";
700 print#1,tab(37)"-"lp"-"ec$"F":lp=lp+1
710 print#1:print#1
720 on bt goto 750,730,740
730 print#1,ec$"M";:goto 750
740 print#1,chr$(15);
750 if ci=0 or lp>1 then 830
760 if em then print#1,ec$"E";
770 print#1,chr$(13)tab(lm(bt)+lt);
780 if al then print#1,ec$"-"chr$(1);
790 print#1,chr$(d);c$;chr$(20)chr$(13):lp=lp+1
800 if al then print#1,ec$"-"chr$(0);
810 if em then print#1,ec$"F";
820 if sk then for l=1 to sk:print#1:next l
830 return
840 print"█     █ A t┌rban m┼g nincs sz▬veg !!! "
850 gosub 530:print"█     Vissza a f▬men┐h▬z:< F-1 >"
860 get w$:if w$<>chr$(133) then 860
870 goto 970
880 me$(1)="Sz▬vegbe▬┌s."
890 me$(2)="Sz▬vegellen▬rz┼s."
900 me$(3)="Form┌tum ellen▬rz┼s"
910 me$(4)="Nyomtat┌s."
920 me$(5)="Lemez katal▬gus."
930 me$(6)="Sz▬veg beh▬v┌s."
940 me$(7)="Lemezrement┼s."
950 me$(8)="File t▬rl┼s"
960 me$(9)="V ▪ G E":q=peek(2):poke 2,0:if q then 1090
970 print"█:hb=0:ke=2:q=0:v=11
980 for l=1 to 9:f=ke:gosub 410:print me$(l):
    ke=ke+2:next l
990 print spc(8)"█ <CRSR-le/fel> <RETURN>"
1000 gosub 530:ke=2:v=9:poke 650,64:for l=1 to 9
1010 f=ke:gosub 410:print ">"
1020 wait 198,1:get w$
1030 if w$=chr$(13) then q=l:l=9:goto 1080
1040 if w$=chr$(17) then ke=ke+2:goto 1070
1050 if l>1 and w$=chr$(145) then ke=ke-2:l=l-2:goto 1070
1060 goto 1020
1070 gosub 410:print" "
1080 next l:poke 650,0:if q=0 then 1000
1090 v=0:uz=q:on q goto 1100,1650,2470,2600,3000,
     3010,3070,3160,3220
1100 print"█:if s=0 then 1160
1110 print"█     █ A t┌rban sz▬veg van ! █
1120 print"█     T▬r▬ljek ?";:gosub 440
1130 if w$="i" then f=5:gosub 410:poke 2,q:goto 3230
1140 poke 2,0:if q>1 then return
1150 goto 1200
1160 gosub 530:n$="":print"█     A file neve:";:input n$
1170 if n$="" or len(n$)>16 then 1160
1180 if q=7 and n$="*" then n$=fi$:f=3:v=16:gosub 410:
     print" "n$:v=0:goto 1200
1190 if n$="*" then 1160
1200 fi$=n$:if q>1 then return
1210 if s then 1480
1220 print"█     H┌ny has┌bot k┼r ? "
1230 get w$:bt=val(w$):if bt<1 or bt>3 then 1230
1240 sh=sh(bt):print"█";
1250 print"   A has┌bok sz┌ma:";bt
1260 print"█     Bet┐t▬pus     ";bt$(bt)
```

1. EPSON dekódolás a kódtáblával.
   Az áthelyezett és módosított BSOUT rutin. Konvertál a BASIC kódról az EPSON (P. C.) kódra.
2. DIR.
   Egyszerre csak 23 sort ír ki.
3. GET rutin.
   Működését lásd fentebb!
4. Sorkiegyenlítés.
5. Hangjelzés generálása.
6. CTRL módosítás.
   A módosított CTRL rutin és kódtábla betöltése a szalagpufferba.
7. Printer előkészítés.
   A nyomtató saját RAM-jába másolt karakterkészletét módosítja az ékezetes karaktereknek megfelelően.
8. Beszúrás/sortörlés.
9. Karaktergenerátor másoló.
   A C-64 karakterkészletét másolja a ROM-ból a RAM-ba. Mivel a program szövegírásra készült, csak a kisbetűs/nagybetűs rész kerül átmásolásra és módosításra.
10. Ékezetes karaktertöltő.
11. A rendszerváltozók átírása.

Itt kell megjegyezni, hogy a gépi kódú programot az első futtatás előtt kell lemezre menteni, mert több helyütt átírja önmagát.

(Nem tudom, hogy terjedelme miatt közlésre kerülhet-e a BASIC betöltő. De az az érdekessége, hogy valamivel kevesebb mint 1700 adat 9 blokkban foglal helyet. Adathiba esetén a program kiírja, hogy a hibát melyik blokkba kell keresni. Így — esetenként — ,,csak" 200 adatot kell átböngészni.)

Mint már említettem volt, fontos szempontnak tekintettem a program kényelmes és könnyű használhatóságát. Persze tudva azt, hogy egy profi szövegszerkesztő ,,komfortfokozatát" meg sem közelíthetem. Így hát nem különcködés, hanem a fentiek miatt alkalmaztam eltérő (Mitől is tér el? Melyik szabványtól?) ékezetes billentyű kiosztást. Íme:

	A	E	R	T	Y	U	I	D	P	@
C-	á	é	—	ű	ü	ú	í	ó	ö	ő
CTRL	Á	É	—	Ű	Ü	Ú	Í	Ó	Ö	Ő

,,Őrültség, de van benne rendszer." (Polonius után szabadon.)

Egyetlen pillantást vetve a klaviatúrára azonnal feltűnik, hogy az ,,á" kivételével minden ékezetes billentyű egyetlen sorban található, az ,,ű"-től pedig sorfolytonosan. És ami talán még ennél is lényegesebb, egyetlen ,,használható" karakter sem veszett el.

A további komfort jegyében a prog-

```
1270 print"■   A max. oldalsz m:";6/bt
1280 print"■   Sor/oldal         :";59*bt
1290 print"■   Karakter/sor      :";sh
1300 print:gosub 430:if w$="n" then 1220
1310 if q=4 then return
1320 gosub 530:print"■   Sorkiegyenl■t-s";:gosub 440
1330 eg=0:if w$="i" then eg=1
1340 poke 51853,eg:poke 51701,sh:
     poke 249,sh-6:if s then return
1350 gosub 530:ci=0:sk=0
1360 print"■   Az irat c■me:■:sys 51702:print" "
1370 x=peek(250):if x=0 then c$="":goto 1400
1380 print"■:gosub 430:if w$="n" then 1350
1390 c$=a$(0):ci=1
1400 if s=r or x=0 then 1470
1410 gosub 530:print"■   K■rem jel■lje ki a c■m ut ni■
1420 print"    ■■res  sorokat! ■
1430 print"■c$"■:print u$
1440 gosub 150:if fu then 1470
1450 get w$:if w$<>chr$(13) and w$<>chr$(133) then 1450
1460 if w$=chr$(13) then print u$:sk=sk+1:goto 1440
1470 t$(0)=c$:if q>1 then return
1480 print "■:if s then gosub 1320:goto 1500
1490 print "■ A sz■veg: ■:goto 1510
1500 print "■ Folytat s a"str$(s+1)". sort■l. ■
1510 if s=0 and ci=0 then 1560
1520 y=s-5:if y<1 then y=1
1530 if ci and y=1 then y=0
1540 hi=s+3:for i=y to s:gosub 320:next i
1550 gosub 150:if fu then 970
1560 sys 51702:print" ■;
1570 if peek(51698)=133 then 1620
1580 x=peek(250):if x=0 then a$(0)=p$:print u$;
1590 s=s+1:t$(s)=a$(0):if x=40 then 1550
1600 if x=39 then print" ";:goto 1550
1610 print:goto 1550
1620 f=peek(214):gosub 420
1630 if w$="n" then gosub 400:goto 1550
1640 goto 970
1650 if s=0 then 840
1660 hi=s+3:uz=2:gosub 530:print"■
1670 gosub 530:print"■
1680 print"■   A szabad  sorok sz ma    :"r-s-(sk*bt)
1690 print"■   A foglalt sorok sz ma    :"s
1700 print"■   H nyadik  sort■l         ";
1710 input w$:y=val(w$):if y=0 or y>s then 970
1720 print"■;:a=1:if y>1 then 1740
1730 if ci then a=0:y=0
1740 gosub 270
1750 for i=a to z:if i<y then 1840
1760 gosub 320
1770 get w$:if w$=chr$(17) then 1820
1780 if w$=chr$(145) then i=i-1:goto 1760
1790 if w$=chr$(136) then hb=hb+1:hb(hb)=i:goto 1820
1800 if w$=chr$(133) then i=z:goto 1870
1810 goto 1770
1820 if i=0 then gosub 380
1830 if hb=25 then i=z:goto 1830
1840 if i<>hv then 1870
1850 hv=hv+m:p=p+1:if i=>y then print hv$
1860 if p=bt then gosub 290
1870 next i:if hb=0 then 850
1880 gosub 1320:print "■    Jav■t rskor is ??";
1890 jv=0:gosub 440:if w$="i" then jv=1
1900 for l=1 to hb:hi=hb(l)
1910 print "■ A m■dos■tand■ sor sz ma:"str$(hi)"
     ■:gosub 2420
1920 for i=y to x:gosub 320
1930 if i=0 then gosub 380
1940 next i:f=peek(214)+2
1950 gosub 400:print spc(12)"Jav■t rs      [1]
```

ram vezérlése kétféle menütechnikával, és az Igen/Nem alternatívával történik.

Ugyancsak a komfort jegyében választottam a nyomtató Epson FX 80-as emulációját. Sokoldalú szolgáltatásai ellenére könnyen programozható és a RAM-ba másolt teljes karaktergenerátor módosítható. (Szemben a többi üzemmóddal, amelyek csak az ún. ,,normál karakterek", tehát a 33 és 126 közötti ASCII kódtartományba eső karakterek módosítását engedik meg. Ergo ROM-RAM átkapcsolgatás nélkül nem lehetne ékezetes karatereket nyomtatni. Persze ezt is meg lehetne oldani, de hát a programozó is ember...)

A lapra nyomtatás lehetőségét is figyelembe véve, egy oldalra 62 sor nyomtatható. Ebből 3 sort foglal le a lapszámozás, vagy a cím. A fennmaradó 59 nyomtatott sorba kerül a választott hasábszámtól függően az 59, a 118, illetve a 177 szövegsor. Mivel a program 354 sor bevitelét engedi meg, a maximálisan nyomtatható oldalak száma rendre: 6; 3; 2.

Az irat címe (amely levél esetén lehet a megszólítás) után kijelölt minden sorköz csökkenti úgy a beírható, mint a nyomtatható szövegsorok számát.

Ha az elkészítendő irat hosszabb mint a hasábszámhoz engedélyezett oldalszám, az iratot újabb file-ban kell folytatni, de most már cím nélkül és a folytatás kezdő lapszámát a nyomtatáskor kell kijelölni.

III. A program behívása és a főmenü

A betöltés és az indítás ,,hagyományos" módon történik. A gépi kódú programot a BASIC hívja be. A töltés kezdetekor megváltozik a képernyő és a karakter szín, ″a'la HELP+″. (Kellemesebb színkombinációt nem találtam.) Színeváltozásával, valamit a THREE-CLEF TEXT felirattal jelentkezik be a program.

A betöltés és a gépi kódú rutinok futásának befejeztével kapjuk a főmenüt, az alábbi üzemmód választékával:

Szövegbeírás.
Szövegellenőrzés.
Formátum ellenőrzés.
Nyomtatás.
Lemezkatalógus.
Szövegbetöltés.
Lemezrementés.
File törlés.
VÉGE.

⟨CRSR-le/fel⟩ ⟨RETURN⟩

A legfelső sor előtt egy ,,jobbra nyíl" látható, amelyet a jelzett billentyűkkel tu-

```
1960 print spc(12)"Besz▔r rs     [2]
1970 print spc(12)"Sort▁rl┴s     [3]
1980 print spc(12)"M A R A D ! [4]■
1990 print spc(12)"V rlasszon    [ ]■■■;
2000 get w$:co=val(w$):if co<1 or co>4 then 2000
2010 gosub 400:on co-1 goto 2100,2150,2410
2020 if jv then 2350
2030 q$=t$(hi):if q$=p$ then q$=""
2032 print "   ■ ',′=<C=+£>  *  ':′=<SHIFT+£>"
2034 print "   ■ ';′=<C=+S>  *  'ト′=<C=+Q> ■■
2040 f=peek(214):print q$:gosub 410
2050 poke 19,1:input q$:poke 19,0
2060 if q$="" then q$=p$:goto 2090
2070 h=len(q$):if h<=sh then 2090
2080 q$=left$(q$+pu$,sh)
2090 t$(hi)=q$:print:goto 2410
2100 if ci or hi>1 then 2130
2110 print "  A c▬met akarja p▬tolni ";:
     gosub 440:gosub 400
2120 if w$="i" then gosub 1350:goto 2410
2130 gosub 150:if fu then 2410
2140 goto 2160
2150 if hi=0 then ci=0:sk=0:goto 2410
2160 k(0)=peek(47)+peek(48)*256+20
2170 if co=3 then goto 2210
2180 k(7)=k(0)+(hi*3)
2190 k(8)=k(0)+s*3+2:k(9)=k(8)+3
2200 poke 2,0:s=s+1:goto 2240
2210 k(7)=k(0)+s*3+3:k(9)=k(0)+hi*3
2230 k(8)=k(9)+3:poke 2,1:s=s-1
2240 k(2)=int(k(7)/256):k(1)=k(7)-k(2)*256
2250 k(4)=int(k(8)/256):k(3)=k(8)-k(4)*256
2260 k(6)=int(k(9)/256):k(5)=k(9)-k(6)*256
2270 poke 247,k(1):poke 248,k(2)
2280 poke 52537,k(3):poke 52538,k(4)
2290 poke 52540,k(5):poke 52541,k(6)
2300 sys 52534
2310 for i=1+1 to hb:if co=3 then 2330
2320 hb(i)=hb(i)+1:goto 2340
2330 hb(i)=hb(i)-1
2340 next i:if co=3 then 2410
2350 t$(hi)="":gosub 400:if jv then 2380
2360 gosub 2420:if bt=1 and x<s then x=x-1
2370 for i=y to x:gosub 320:next i
2380 gosub 410:sys 51702:poke 198,0
2390 if peek(250)=0 then t$(hi)=p$:goto 2410
2400 t$(hi)=a$(0)
2410 next l:goto 970
2420 y=hi-2:if y<1 then y=1
2430 if ci and y=1 then y=0
2440 x=hi+2:if x>s then x=s
2450 return
2470 if s=0 then 840
2480 hi=s+3:lp=1:d$="     ":gosub 270:print"■
2490 if ci=0 then print spc(17)"■■-"lp"-■■:goto 2510
2500 print left$(c$,39):gosub 380:lp=lp+1
2510 for l=1 to z:w=l:pr$="":gosub 200
2520 get w$:if w$<>chr$(17) and w$<>chr$(133) then 2520
2530 if w$=chr$(17) then 2550
2540 l=hv:w=l+(bt-1)*m
2550 if w=z then l=z:goto 2590
2560 if l<hv then 2590
2570 print spc(17)"■■-"lp"-■■:lp=lp+1
2580 i=w:gosub 290:l=w
2590 next l:print"■   ";:goto 850
2600 if s=0 then 840
2610 gosub 540:if hb then 970
2620 if bt=1 then 2640 .
2630 co=bt:gosub 1240:if bt<>co then uz=3:goto 2480
```

```
2640 lp=1:print"■:if ci then 2690
2650 input"■■    A kezd_ lapsz m: ";w$
2660 lp=val(w$):if lp<1 then 2640
2670 print"■:gosub 430:if w$="n" then 2640
2680 goto 2820
2690 f=6:v=25:print"■■■    A c■met:■
2700 h=len(c$):if bt=1 and h>32 then 2720
2710 d=20:print"■   Duplasz+less+ggel ? ";:gosub 470:if w$="i" then d=14
2720 ko=0:print"■   K_z+pre ?           ";:gosub 470:if w$="i" then ko=1
2730 al=0:print"■   Al h"zva ?          ";:gosub 470:if w$="i" then al=1
2740 em=0:print"■   Kiemelt nyomtat rs ? ";:gosub 470:if w$="i" then em=1
2750 lt=0:if ko=0 then 2810
2760 if d=14 then h=h*2
2770 on bt-1 goto 2790,2800
2780 lt=int((64-h)/2):goto 2810
2790 lt=int((84-h)/2):goto 2810
2800 lt=int((128-h)/2)
2810 v=0:print "■■;:gosub 430:if w$="n" then 2690
2820 fo=1:gosub 270
2830 print"■■■:if z<=(59-sk)*bt then 2860
2840 print"    ■ Folyamatos nyomtat rs ? ■;:gosub 440:print"■■■
2850 if w$="i" then print"     Leporell■ nyomtat rs !":goto 2870
2860 fo=0:print"     Egy lap nyomtat rsa !"
2870 f=7:lm$="":for a=1 to lm(bt):lm$=lm$+" ":next a:h=sh:d$="     "
2880 gosub 670:for l=1 to z:pr$=lm$:w=1
2890 gosub 200
2900 if w=z then l=z:goto 2980
2910 if l<hv then 2980
2920 print#1,chr$(12);:i=w:gosub 290:l=w
2930 if fo then 2970
2940 gosub 400:gosub 530:print"      K+rem cser+lje ki a lapot,■
2950 print"    majd nyomja meg a HLOCAL+-t !■■:gosub 430:if w$="i" then 2970
2960 gosub 420:if w$="i" then l=z:goto 2980
2970 gosub 400:gosub 680
2980 next l:print#1,chr$(12);
2990 print#1,ec$"P"chr$(18);:goto 970
3000 sys 51497:poke 198,0:print"  ";:goto 850
3010 if s then gosub 1100:goto 970
3020 gosub 1160:open2,8,2,n$+",s,r":gosub 510:if e then 850
3030 input#2,fi$,bt,sk,c$:sh=sh(bt)
3040 s=s+1:input#2,t$(s):if st=0 then 3040
3050 ci=0:if len(c$) then ci=1:t$(0)=c$
3060 close2:close15:goto 1660
3070 if s=0 then 840
3080 gosub 1160:open2,8,2,n$+",s,w":gosub 510:if e=0 then 3130
3090 if e<>63 then 850
3100 print"■■    Fel"l■rjam ? ";:gosub 440:if w$="n" then 970
3110 gosub 420:if w$="n" then 970
3120 open2,8,2,"@:"+n$+",s,w"
3130 print#2,fi$;",";bt;",";sk;",";c$
3140 for l=1 to s:print#2,t$(l)
3150 next l:close2:close15:goto 3210
3160 gosub 1160:if hb then 970
3170 open2,8,2,n$+",s,r":gosub 510:close2:if e then 850
3180 gosub 420:if w$="n" then 3200
3190 print#15,"s:"+n$
3200 close15:if w$="n" then 970
3210 gosub 530:q=5:goto 1090
3220 print"■■;
3230 gosub 420:if w$="i" then 3260
3240 poke 2,0:if q=1 then return
3250 goto 970
3260 if q=9 then sys 64738
3270 clr:goto 90
```

dunk mozgatni a kívánt üzemmódra. A ⟨RETURN⟩ után bejelentkezik a választott üzemmód promtja. Ez lehet egy almenü, de lehet egy hibaüzenet is. Két kivétel van: A DIR rutin azonnal aktivizálódik, a VÉGE esetében pedig visszakérdez: „Egészen biztos benne?”

Ez a „szkeptikus magatartás” jellemzője a programnak, és a már említett „jóvátehetetlen” következményekkel járó utasítások kivédését célozza. Pl. a tár, vagy egy file törlése, felülírása. Ugyanis ha „nem”-mel válaszolunk, általában a főmenüt kapjuk vissza. Ellenkező esetben az utasítás végrehajtódik. A VÉGE esetében a sys 64738-as ROM rutin. (RESET)

A főmenü bejelentkezésekor a vezérlőbillentyűk ismétlése letiltásra kerül, tehát „egy lenyomás — egy lépés”.

A továbbiakban az üzemmódok részletes ismertetése következik, a főmenü sorrendjében.

SZÖVEGBEÍRÁS

Három alapeset lehetséges:

1. Az összes szövegstring „üres”.

Promt-ként egy kérdést kapunk: „A file neve?”. Azt a nevet kell beírni, amely néven a lemezen fog szerepelni. Üres stringet nem fogad el.

A file nevének megadása után választhatjuk ki 1-től 3-ig a nyomtatandó hasábok számát. Ezzel egyúttal a betűtípust is kiválasztottuk. Ezután egy táblázatot kapunk a nyomtatás főbb jellemzőivel:

A hasábok száma:	1	2	3
A betűtípus:	PICA/ELIT/CONDENSED		
Az oldalak száma:	6	3	2
Karakter/sor:	64	40	40 [HS]

O.K. ? ? ?

Nemleges válasz esetén módosíthatjuk a nyomtatható hasábok számát. (Természetesen, csak a választott hasábszámnak megfelelő oszlop adatai íródnak ki a képernyőre.)

Ezután az irat címét kéri a program. Ha írunk címet, a cím utáni üres sorok kijelölése következik. Minden ⟨RETURN⟩ egy üres sort eredményez. Befejezni a sorközi kijelölését az ⟨F-1⟩-gyel lehet.

Ezután következik az utolsó kérdés: „Sorkiegyenlítés?”. Ha „igen”-nel válaszolunk, aktiválódik a sorkiegyenlítő rutin. (Sorkiegyenlítés történik, ha a beírt sor hosszabb mint SH-6 és a sor 2-nél több szóközt tartalmaz. A rutin, működése során az első szóköztől kezdve addig „szúr be” újabb szóközöket, amíg a sorhossz el nem éri az SH értékét.) A válasz után megjelenik a villogó kurzor, és a program karakterbevitelre vár.

```
0 rem "TCT basic betolto program"
10 :
20 ck=51200:f=145
30 print "■:read v$:if v$="@" then 110
40 s=peek(63)+peek(64)*256
50 read v:cn=0:for l=1 to len(v$) step 2
60 d=(asc(mid$(v$,l,1))-65)*16+asc(mid$
   (v$,l+1,1))-65:cn=cn+d
70 print " poke"str$(ck+c)","mid$(str$(d),2,3)
80 poke(ck+c),d:c=c+1:next l
90 if v=cn then 30
100 goto 200
110 print "■■ ■ O.K. O.K. O.K. ":cv=ck+c-1
120 print "■■ A programlemez O.K. ?? ■I■gen/■N■em"
130 get w$:if w$<>"i" and w$<>"n" then 130
140 open15,8,15,"i"
150 x=int(ck/256):y=ck-x*256
160 open2,8,2,"gepi/tct,p,w":print#2,chr$(y);
    chr$(x);:for l=ck to cv
170 print#2,chr$(peek(l));:next l
180 close2:close15
190 print "■■ ■ O.K. O.K. O.K. !! ■:end
200 print"■■■ ■ A D A T H I B A !! ■■
210 print "list"str$(s)
220 print"■■■run":poke 198,8
230 for l=631 to 637:poke l,f:next l
240 poke l,13:end
250 :
300 data emfdmleikfjkmjadnaaegiembgohjabg,1768
310 data kflkmjaenaamieacgikiljcjmikeacem,1850
320 data cdmigiemnnonemnlpbaaabacadaeafag,1430
330 data ahaiajanalamakaoapaqbbbcbdbebfbg,232
340 data bhbibjbkblbmbnbobpcacbcccdcecfcg,488
350 data chcicjckclcmcncocpdadbdcdddedfdg,744
360 data dhdidjdkdldmdndodpeagbgcgdgegfgg,1192
370 data ghgigjgkglgmgngogphahbhchdhehfhg,1768
380 data hhhihjhkflfmfnfofpgagbgcgdgegfgg,1640
390 data ghgigjgkglgmgngogphahbhchdhehfhg,1768
400 data hhhihjhkhlhmhnhohpiaibicidieifig,2024
410 data ihiiijikiliminioipjajbjcjdjejfjg,2280
420 data jhjijjjkjljmjnjojpkakbkckdkekfkg,2536
430 data khkikjkkklkmknkokplalblcldlelflg,2792
440 data lhliljlklllmlnlolpmaebecedeeefeg,2280
450 data eheiejekelemeneoepfafbfcfdfefffg,1256
460 data fhfifjfknlnmnnnonpoaobocodoeofog,3048
470 data ohoiojokolomonooppapbpcpdpepfpg,3816
480 data phpipjpkplpmpnpoppkjjdcancppkjce,3277
490 data ifpnkjpnifllkjaaiflmkjabiflhkjai,2281
500 data iflkkjgaifljcanfpdkflkcaleppkflj,2558
510 data cajgppkjaainpdmjifjakaadiepncakf,2213
520 data ppifpokejanaehcakfppkejanaeakepn,2678
530 data iinaojkgpocamnlnkjcacancppcakfpp,2573
540 data kgjanackkkpaagcancppemhnmjkjanca,2089
550 data ncppkmpdmjmabgnaaokaaaimpdmjkeml,2628
560 data maabnapkcaeeofoopdmjkaacnakocaec,2304
570 data pggaaaaaaaaaaaaaaaaaaaaaaaaaaaaa,342
580 data aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa,0
590 data aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa,0
600 data aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa,0
610 data aaaaaakaaaacaclmlkcaaigmgigmmkf,1285
620 data mgmjaapapkknhhacmjcalaademfjmkmj,2163
630 data injaahmjkalaademgomkmjifpafhmjcc,2116
640 data naafkjklemdpmkmjcmnaafkjkiemdpmk,2030
650 data mjdknaafkjkjemdpmkmjdlnaackjkojn,2121
660 data lcmjoicamkpbkjaaifmgompfmjpacgoe,2774
670 data pjnakmcaclmlemppmjmjbenaanoaaana,2313
680 data afigmgemppmjmkemedmkmjanpaahkjaa,2046
690 data ifmgemppmjinpcmjiopemjigpkkjabmf,2785
700 data mppapmifmmoaaanaabgakjaamjaapahl,2298
710 data kaaaimpcmjmmpemjpaaoljlcmjmjcana,2651
```

Mint már a bevezetőben említettem, nincsenek ,,tiltott karakterek". Csupán egy dologra kell ügyelni: ha a sor szóközzel kezdődik (pl. bekezdés vagy felsorolás) a szóközt ⟨SHIFT⟩+⟨SPACE⟩ [CHR$(160)] kombinációval hozzuk létre. Két oka is van ennek:

a) Ha a stringeket "input # " utasítással olvassuk vissza lemezről, akkor a stringek elején levő CHR$(32) kódok ,,lenyelődnek". Emiatt akár képernyőn, akár nyomtatón jelenítjük meg a szöveget, az csak az első ,,hasznos" karakterrel kezdődik.

b) A sorkiegyenlítő rutin a CHR$(32) kódokat keresi, szükség esetén ezek mellé teszi a kiegészítő szóközöket.

Bármelyik ok önmagában is elronthatja a formátumot.

A fentiek miatt, ha a sorban olyan szópár fordul elő, amelynek széthúzása rontaná a sor esztétikus megjelenését, vagy értelemzavaró lenne, akkor is az említett módszert alkalmazzuk.

A sor írásakor az SH-6 sorhosszat elérve hangjelzést kapunk.

Egy szövegsort vagy a ⟨RETURN⟩-nel, vagy a sor végigírásával tudunk befejezni. Ez utóbbi esetben hangjelzés figyelmeztet az új sor megkezdésére. (NB. Minden üzenetre hangjelzés figyelmeztet!) Csak a szövegsor befejezése előtt tudunk visszalépni az ⟨INST/DEL⟩ billentyűvel.

Az üzemmódból való kilépési szándékunkat az ⟨F-1⟩-gyel tudjuk jelezni. A válasz egy kérdés: ,,Egészen biztos benne?" Nemleges válasz esetén ott folytathatjuk, ahol abbahagytuk, egyébként a főmenü tér vissza.

2. A memória megtelt.

Hibaüzenetet kapunk: ,,A memóriában nincs több hely! Töröljek?" Igenlő válaszra a szokásos visszakérdezés: ,,Egészen biztos...?" Az újabb igenre végrehajtódik egy CLR, a tömbök újradimenzionálása, a változók visszaállítása. Ezután az 1. alapeset szerinti módon folytatódik a program.

,,Nem" válasz után természetesen a főmenüt kapjuk vissza.

3. A tárban szöveg van.

Ez esetben a fenti üzenetet kapjuk, a Töröljek? kérdéssel kiegészítve. ,,Igen" válasz esetén a 2. alapesetnek megfelelően folytatódik a program. A ,,Nem"-et követő kérdésre (Sorkiegyenlítés?) válaszolva üzenetet kapunk: ,,Folytatva az XXX. sortól." Ezután kiíródik a tárban levő szöveg utolsó 5 sora, a következő sorban a villogó kurzorral.

```
720 data adoopcmjmiemjfmkkopcmjoaaapafmkn,2657
730 data pfmjdionpemjmjaglafboaabnaaemjad,2289
740 data laejkmpemjkcaaompemjpadhlnlcmjoi,2804
750 data mjcanapdiopdmjmopdmjmmpfmjpacmmm,3058
760 data pdmjpaamljlcmjmijjlcmjiiiiemnpmk,2765
770 data kjkakopdmjoijnlcmjoioopemjkmpemj,3247
780 data emmhmkknpfmjmnpemjnalhkfcpkgdabi,2587
790 data gjahjaaboiifphigpikaaaknpemjjbph,2421
800 data mikjlcjbphmikjmjjbphgakjaakijjaa,2487
810 data nemaakpaaemiemcomlkjalkaapinafne,1896
820 data imbinekjbokajpinapneimabnfkjbnin,1955
830 data aenegakcaalngdmljndmadoioagdnapf,2193
840 data emiamnemfeadkninacmjaepaademelol,1722
850 data kjfpifpfkjadifpgemoaokkjdpinipac,2245
860 data kjadinjaacgapppppppppppppppppbmbh,2646
870 data knjpbkbdkmppjmbcaeboadagkfbibple,1421
880 data ahjoacailfbgbckbakjcanallgaoppkk,1358
890 data ampppbblkhppbmppbnppppbpboppjaag,2259
900 data ppafppppbbppkjabkcaekaaacaaapokj,1993
910 data aacabipocaekpdkcabcafapckjpjkcml,1959
920 data ifphigpikaaalbphmjpppaamcannonog,2774
930 data phnapbogpiemoemlgabibldkaaaaaabl,1913
940 data cgaakbljiaaaaaicaapoaaiccaeaiaaa,1250
950 data iaaaccaadoeaicaaaaaaaaaaiadmacea,672
960 data icaaeciadmacaaaaiabmccaagciaccea,900
970 data kcbmaaaaiahmacaamcaamcaaachmaaaa,958
980 data iaaabmccaaccbeaibeccaaaaiadmecaa,560
990 data ocaaocaaecdmaaaaaaaaaaaamabaoaaaa,605
1000 data aaaaaaaaiaaaaadgaadgaaaaaaaaaaaa,236
1010 data iadmeciaecaaeciaecdmaaaaiaaaaaoa,1120
1020 data aaaaaaoaaaaaaaaaiadoaackaackeakk,732
1030 data aaccaaaaiaaoaqcieaiieacifaioaaaa,758
1040 data aaaaaaaagmabgoaaaaaaaaaaaiabmccia,537
1050 data ccaacciaccbmaaaaiaaeakcaakgaikca,708
1060 data bmacaaaaiabmccaicceikcaiccbiaaaa,562
1070 data iaaaaaaaaaaaaaaaaaaaaaaiaaaaaaa,256
1080 data aaaaaaaaaaaaaaaaiahmacaaicaaicaa,514
1090 data achmaaaaiapmacaaacmaacaaacpmaaaa,958
1100 data iadmecaaeckaecaaecdmaaaaiadmacia,990
1110 data acaaaciadmacaaaaiadmacaaeciaacaa,580
1120 data dmacaaaaiabmccaacceakcaaccbmaaaa,574
1130 data blcfabaabbppkgacknllkkinnnmmkndj,1831
1140 data mnmfphnaalkndkmnmfpinaaecacolfga,2316
1150 data oaaanabjkndjmnnaadmodkmnmodjmnkn,2213
1160 data dmmnnaadmodnmnmodmmnemdimnoodjmn,2256
1170 data naadoodkmnoodmmnnaloooodnmnemdimn,2454
1180 data hikjddifabkcaakaailnaanijnaaoioi,1830
1190 data naphooilmnooiomniinaookjdhifabfi,2650
1200 data kcaalnnimnjnaiolkjppdipnnimnjnai,2491
1210 data opoioaminaomkjadkcmiincgadiochad,2239
1220 data kjmekcbliniiacinaanniobinakjgain,1975
1230 data aamicadmademeeofhjdddcdadadahiaa,1154
1240 data ambiaadibibidmaaddggaaggggggdoaa,721
1250 data ddggaadmggggdmaagdmgaamdmdmdhoaa,1485
1260 data mmmmddddmmmmddddddggaahomdmdhoaa,1815
1270 data aaaaaaaaabibidaaaaabiaaaabiaaaa,144
1280 data ggggaahomdmdhoaaggggggaaaaaaaaaa,1152
1290 data aghogmgahigahoaabldpgghogggggaa,1302
1300 data aaaabiaaaabibidaggggaadmggggdmaa,648
1310 data ambidmagdoggdoaaambidmgghogadmaa,808
1320 data aaaaaappppbibibibibibibipipibibibi,1222
1330 data ggggaamdmdmdhoaabinlnlmdmdmdhoaa,2088
1340 data bihonlmdmdmdhoaaggggaaggggggdoaa,1652
1350 data ambiaaggggggdoaaambiaadmggggdmaa,764
1360 data pp,255,@

ready.
```

SZÖVEGELLENŐRZÉS

Hibaüzenettel is kezdődhet a rutin: ,,A tárban még nincs szöveg!" ,,Vissza a főmenühöz: (F-1)". (A második üzenettel majd' minden üzemmód befejeztével találkozhatunk.) Ha viszont a tárban van szöveg, a pront kiírja a még szabad, valamint a már foglalt szövegsorok számát és választ vár; ,,Hányadik sortól kéri?"

Ha 0-val, vagy karakterrel válaszolunk, a főmenübe térhetünk vissza. Ez jól jöhet akkor, ha csupán a tár foglaltságára vagyunk kíváncsiak. A kiírt maximális sor számánál nagyobb értéket nem fogad el a program. A választ, beírása után (RETURN)-nel kell bevinni.

Ha az első oldaltól kérjük a szöveget, és van címe az iratnak, a címmel kezdődik az irat kiírása. Továbblépni a (CRSR/le) vagy az (F-7) billentyűkkel lehet. Visszalépésre a (CRSR/fel) szolgál.

A sorközök nem üres sorként jelennek meg, hanem az aktuális sorszámmal, megkönnyítve ezzel a szövegben való tájékozódást. A hasábok végét (lapalja) egy speciális elválasztó sor jelzi.

Ha egy sorról az (F-7)-tel lépünk tovább, az a sor módosítandó sorként ,,vétetik nyilvántartásba". Ily módon ,,egy menetben" 25 sort lehet módosításra kijelölni.

Mint minden üzemmódból, ebből is az (F-1)-gyel lehet kilépni. Amennyiben ,,bejegyzés" történt a hibanyilvántartásban, a program azonnal a módosító rutinba lép. Ellenkező esetben a főmenüt kapjuk vissza.

A módosító rutin a ,,Sorkiegyenlítés?" kérdéssel jelentkezik be, majd újabb kérdés következik: ,,Javításkor is?" Válaszunktól függően kétféle módon javíthatunk. ,,Igen" válasz esetén üres sorba, a megszokott módon kell beírni a javított szöveget.

,,Nem" válasz esetén kiíródik a módosítandó sor, és felette a ,,tiltott" karakterek billentyű kombinációi. Ez esetben ugyanis ,,input" utasítással történik a javítás. Annak minden előnyével és hátrányával egyetemben. Választásunk után üzenettel folytatódik a program: ,,A módosítani kívánt sor száma: XXX".

Ezután 5 szövegsor íródik ki, a középső (módosítandó) sor inverzben. Így szövegkörnyezetében és kiemelve láthatjuk a kritikus szövegsort. A sorok alatt az almenü következik:

Javítás [1]
Beszúrás [2]
Sortörlés [3]
MARAD [4]

Választás után törlődik az almenü, majd a választástól függően alakul a program további futása:

[1] A program karakterbevitelre vár.
[2] A ,,helycsinálás" befejeztével kiíródik újabb 5 sor. A kurzor a középső üres sorban várja a beszúrni kívánt szöveget.
[3] A sortörlés befejeztével a program a következő kijelölt sorra lép.
[4] Azonnal tovább lép.

Amennyiben az 1. sort is kijelöltük, az irat címét is javíthatjuk, törölhetjük, ha még hiányzik pótolhatjuk. Az utolsó módosítási művelet után a főmenüt kapjuk vissza.

FORMÁTUMELLENŐRZÉS

Nagyon könnyen előfordulhat, hogy — nem is valószínűtlen példákat említve — egy fejezetcím a hasáb utolsó sorába kerül, vagy netán a hasáb üres sorral kezdődik. Az ilyen jellegű — formátumrontó — hibák nyomtatás előtti felderítésére szolgál ez az üzemmód. Ha még nincs szöveg a tárban, hibaüzenetet kapunk.

A kijelölésnek megfelelően a hasábok első néhány karaktere látható egymás mellett úgy, ahogyan azok a nyomtatott lapon fognak elhelyezkedni. A sorközök ebben az esetben is tartalmazzák az aktuális sorszámot.

A sortovábbítás a (CRSR/le)-vel történik. Az (F-1)-gyel ,,lapozhatunk". Ha elfogyott a szöveg, a főmenü jön vissza.

Jótanács: Mindenekelőtt a kész irat szöveghűségét ellenőrizzük, és csak a hibák kijavítása után vizsgáljuk a formátumot. Az észlelt ,,anomáliák" sorszámát jegyezzük fel. Ezután sorközök törlésével vagy beszúrásával végezzük el a ,,fazonigazítást".

NYOMTATÁS

Hibaüzenetet is kaphatunk: ,,A tárban még nincs szöveg!" Ha van szöveg és még nem nyomtattunk, a printer aktivizálása következik. A további két alapeset lehetséges:

1. Az iratnak nincsen címe.
   Ekkor a kezdő lapszámot kéri a program. Ez csupán látszólag értelmetlen kérés. Ha ugyanis 354 sornál hosszabb iratot készítünk, akkor a folytatást új file-ba, cím nélkül kell kezdeni. A folyamatos lapszámozás csak így biztosítható.
2. A cím ,,tipográfiájára" vonatkozó kérdéseket kapjuk:
   Duplaszélességgel?
   Középre?
   Aláhúzva?
   Kiemelt nyomtatás?

A felkínált lehetőségeket tetszőleges kombinációban, — akár valamennyit együtt — választhatjuk. Újabb kérdés következik: ,,Folyamatos nyomtatás? Igen/Nem".

A kérdésre adott választól függően üzenetet kapunk:
(Igen) ,,Leporelló nyomtatás."
(Nem) ,,Egy lap nyomtatása!"
Ezután kezd dolgozni a printer.

A lapnyomtatás végén (a papírvégérzékelő AKTÍV!) tájékoztatást kapunk a folytatás mikéntjéről.

LEMEZKATALÓGUS

,,No comment!"

SZÖVEGBETÖLTÉS

Ha a tárban szöveg van, üzenetet kapunk és egy kérdést a tárban levő szöveg törlésére vonatkozóan. Igenlő válasz esetén törlődik a tár. A file nevében használhatunk "*"-ot. Sikeres betöltés után a szövegellenőrző rutin jelentkezik. DOS hibaüzenet után pedig: „Vissza a főmenühöz."

LEMEZREMENTÉS

Ha nincs szöveg a tárban, üzenetet kapunk, és vissza a főmenühöz. Egyébként: A file nevét kéri. Ha "*"-ot írunk be, akkor a szövegbeírás kezdetén megadott, vagy a file-ból kiolvasott néven menti lemezre a szöveget. Természetesen új nevet is adhatunk.

Ha a file már létezik, a ,,file exists" üzenet után kérdést kapunk: ,,Felülírjam?" Nemleges válasz esetén: ,,Vissza a főmenühöz." Sikeres lemezművelet után a DIR-rutin következik.

FILE TÖRLÉS

A file nevét kéri. Megvizsgálja a file meglétét és típusát. Ha mindezt rendbenlevőnek találja: ,,Egészen biztos benne?" A választól függően vagy végrehajtás következik, vagy: Vissza a főmenühöz!

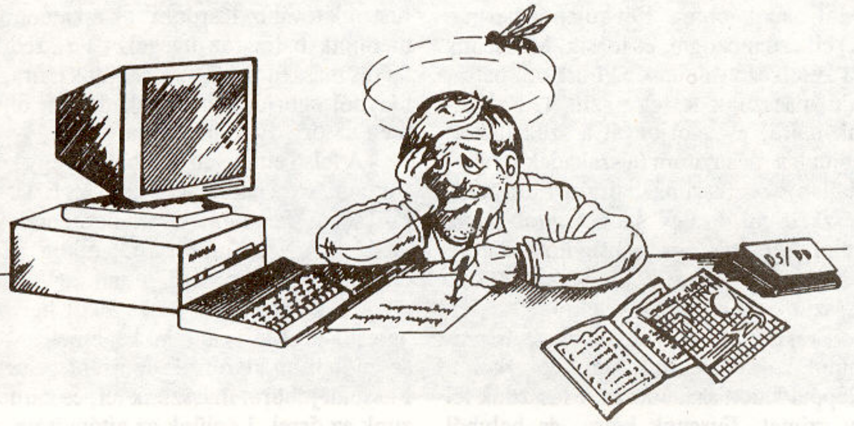
VÉGE

,,Egészen biztos benne?": Nemleges válasz esetén: ,,Vissza a főmenühöz!" Egyébként pedig meghívja a RESET ROM-rutint.

Szász Sándor

C 64 & AMIGA ROVAT

BUMBLE BEE & FRIEND TWO CREW



PRINCE OF PERSIA

1990 nyarának termése a Prince of Persia, vagyis Perzsia hercege címet viselő program. Konkrétan az általános kategóriákba nem sorolható be, az akciós és ügyességi játékok keveréke. A program egy lemez terjedelmű, a számítógép memóriájából 1 MB helyet foglal el. Emiatt az Amiga 500-as alap kiszereltségben nem tudja futtatni. Az általában kékes háttérrel rendelkező, grafikailag és zeneileg gyengén kivitelezett játék varázslatossága a logikai gondolatmenetre, gyors helyzetfelismerésre, időérzékre épül. Aki a programot elkezdi, bizonyosan több napot fog majd a számítógép mellett tölteni. Ahhoz, hogy a leírás elkészüljön a játékot apró részletekig boncolgattuk, végeredményben 12 órai videóanalizálás segítségével sikerült.

A program első látásra is kiemelkedő elemet tartalmaz, mégpedig az emberi mozgás élethű animációs feldolgozását. Sajnos a figura irányítása kissé nehézkes. A játék joystick utasításaira csak későn reagál. Ennek köszönhetően sokszor kezdhetjük az aktuális szintet (ahol éppen tartózkodunk) elölről. A trainerezés a program egy igen fontos kelléke, mivel a játék nehézsége miatt, az örök idő és az örök energia nélkül végigjátszhatatlan. A trainer egyszerűen töltés közben a mouse jobb gombjával aktiválható. Az örök energia, bizonyos esetekben csak részleges védelmet nyújt. A program 12+2 szintből áll, a börtön, a palota, a torony részeiből. Nézzük röviden a játék történetét.

A Szultán elutazik. Perzsia ideiglenes uralkodója, Jaffar Főlátnok, aki kemény, kegyetlen diktátor. A trón megszerzésének csak egyetlen akadálya van, mégpedig a Szultán lánya. A gyönyörű lányt választás elé állítja, vagy elveszi feleségül, vagy meghal. Egy órát kap Jaffartól. A hercegnő egyetlen lehetősége szerelme, aki Jaffar jóvoltából börtönbe raboskodik. Az ifjút a játékos személyesíti meg, akinek 60 perce van, hogy kiszabadítsa a lányt.

A botkormánnyal a következőképpen irányíthatjuk a figurát: Ha a joysticket balra húzzuk, az ifjú balra szalad (egyszeri futás két lépésnek felel meg), ellenkező esetben jobbra. Futás közben az első szakasz végéig irányváltás nem lehetséges. A másik lehetőség a lépkedés, amit a tűzgomb és a joy bal, vagy jobb irányú mozgatásával érhetünk el. Ugrani többféleképpen tudunk: helyből, vagyis álló helyzetből a joystick sré irányú mozgatásával, gombbal, vagy gomb nélkül. Nekifutásból is tudunk ugrani, ha a botkormányt balra húzzuk, és futás közben sré irányba mozgatjuk. A tűzgombot itt is célszerű megnyomni, ez nem minden esetben szükséges. Előfordulhat, hogy fel kell csimpaszkodni, mászni egy helyre, ilyenkor egyszerűen fel lehet nyomni a botkormányt. Ha leereszkedünk pl. egy szintet, az ellenkezőt kell tennünk.

Néhány szakadéknál csak úgy tudunk ugrani, ha a tűzgombot nyomva tartjuk. Ellenkező esetben lezuhanunk. Az utolsó pályák (szintek) valamelyikén a szakadékot csak nekifutásból tudjuk átugrani. Leguggolni is egyszerűen lehet, ha a botkormányt magunk felé húzzuk. A tárgyakat is így szedhetjük fel (tárgy pl. a kard, üveg stb.). A kis üvegek életerőt, a nagy korsók maximális életerőt adnak. Néha kardoznunk kell Jaffar embereivel, a kardot a figura ilyenkor automatikusan veszi elő. Az ellenfelünket a következőképpen győzhetjük le: ha megvárjuk míg a figura üt, ekkor a joyt sré (fel) irányba mozgatjuk, az ütést hárítjuk; majd a tűzgomb segítségével csapást mérünk. Ha az ütés talált az alak hátrább lökődik. Mi is lépjünk egyet is üssünk. Nem kell az ellenfél teljes életerejét megsemmisítenünk. Ha szerencsénk van szakadékba, lecsapódó giotin alá, karókkal felszerelt verembe lökhetjük.



A képernyő jobb oldalán alul láthatjuk a katona energiáját. A számítógép klaviatúráját tulajdonképpen nem kell igénybevennünk, mégis néhány bilentyű használható. A CTRL R megnyomásakor a programot az elsőszinttől kezdhetjük. CTRL Q = Quit. A játék SAVE/LOAD funkciót is tartalmaz, ami néhány verzió kivételével a CTRL G-vel hívható elő. Nézzünk néhány olyan lényeges akadályt, ami a programban befolyásolhatja a továbbjutást.

A legnagyobb számban a vasrács fordul elő a szinteken. Majdnem minden helységben található egy, vagy több ajtónyitó kőlap, amire ha rálépünk felhúzódik a rács. Az ajtónyitó, vagy rácsnyitó kőlap mindig kiemelkedik a többi közül. Vannak kőlapok amelyek a rács mellett közvetlenül találhatók, ha erre vagy ezekre rálépünk, a rács lecsukódik. Olyan kőlapokkal is találkozhatunk, amelyekre ha rálépünk lezuhannak. A törmelékek néha egy ajtónyitó, vagy rácszáró kőlapra zuhannak. Az első esetben a rács nem fog lecsukódni. Ilyen lehetőség másképpen is elérhető, ha egy katonát éppen a nyitón teszünk ártalmatlanná. A másik zavaró tényező az aljzatból előbukkanó karók, ha belerohanunk, felnyársalódhatunk. Lépkedéssel, lefékezéssel ez az akadály is kikerülhető. A magasabb helyekről ne nagyon ugráljunk le, ha két emeletet zuhanunk meghalhatunk, vagy karókba csapódhatunk.

A lecsapódó vasfogak nem jelenthetnek nagy akadályt, ha aprókat lépünk és az elemek záródása után próbálunk átjutni. Ha egymás után sok vasfog található, lépéskor csak az aktuális elemre figyeljünk. A plafonon lehetnek olyan kőlapok, amelyeket ha megmozgatunk leesnek. A mozgatáshoz egyszerűen fel kell ugrani. Akadályt jelenthet a leeső kőlap is, ha a fejünkre zuhan. Ezt a következőképpen kerülhetjük ki: a felugrás után (amikor már mozog a kőlap) rögtön guggoljunk le. A keletkezett nyíláson fel lehet mászni. A katonák sokszor okozhatnak bosszúságot is, amikor belelöknek egy karóba, esetleg egy vasfogba. Ellenfeleink lehetnek még csontvázak is.

A negyedik szinten jelenik meg a tükörképünk, aki három szinten is megtréfál. Tulajdonképpen a felmerülő akadályok közül ezek jelenthetnek nagyobb problémát. A játék szintjeit a következőkben részletesen fogjuk leírni, ezért aki a programot végig szeretné játszani, az írást csak segítségként használja. A szintek a következők:

1. Az első képnél vagyunk, ahol balra fennt egy rács, mellettünk két égő fáklya, jobbra alul és felül egy-egy út látható. Induljunk el jobbra, ereszkedjünk le az alsó útra, fussunk jobbra. Két kőlap (a harmadik) elkezd mozogni, és leesik. Másszunk le a keletkezett nyíláson. Fussunk balra, majd másszunk fel egy szintet. Szaladjunk balra, és ugorjuk át a szakadékot. Lépjünk a rácsnyitóra (a szakadék mögötti kőlap), és fussunk balra a rács alatt. Ereszkedjünk le egy szintet, szaladjunk jobbra, lépjünk rá a rácsnyitó kőlapra. Nekifutásból ugorjuk át a szakadékot. Másszunk fel és fussunk balra.

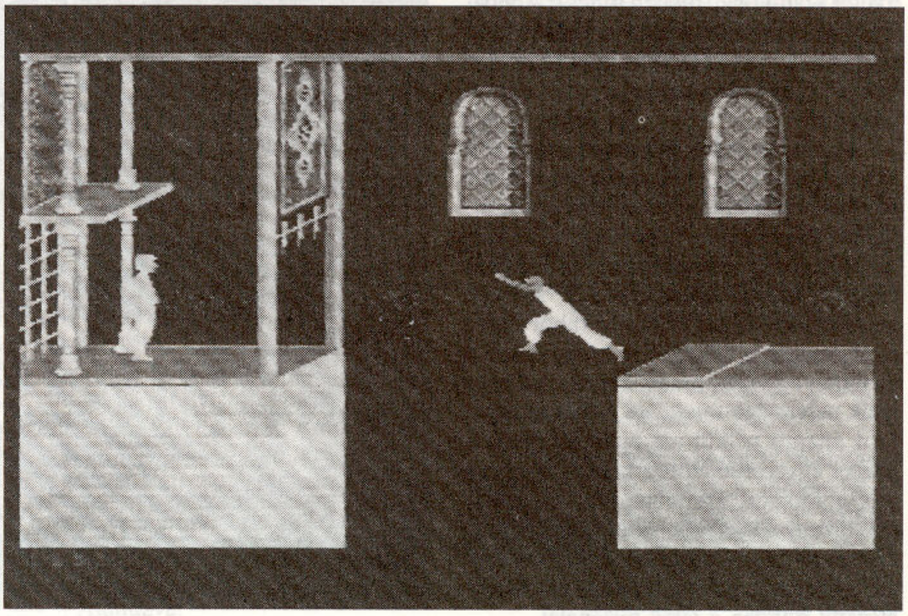
Ereszkedjünk le egy szintet, és szaladjunk balra. Nekifutásból ugorjuk át a kőlappal fedett szakadékot. Másszunk fel egy szintet. Fussunk balra, és helyből gombbal ugorjuk át a szakadékot. Szaladjunk balra és ereszkedjünk le két szintet. A csontváz mellett találhatjuk a kardot. Vegyük fel, és menjünk vissza a második képhez. Fussunk jobbra, és küzdjünk meg az őrrel. Menjünk ismét jobbra, ugorjunk fel a bal oldalon levő kőlapra, az ajtó kinyílik.

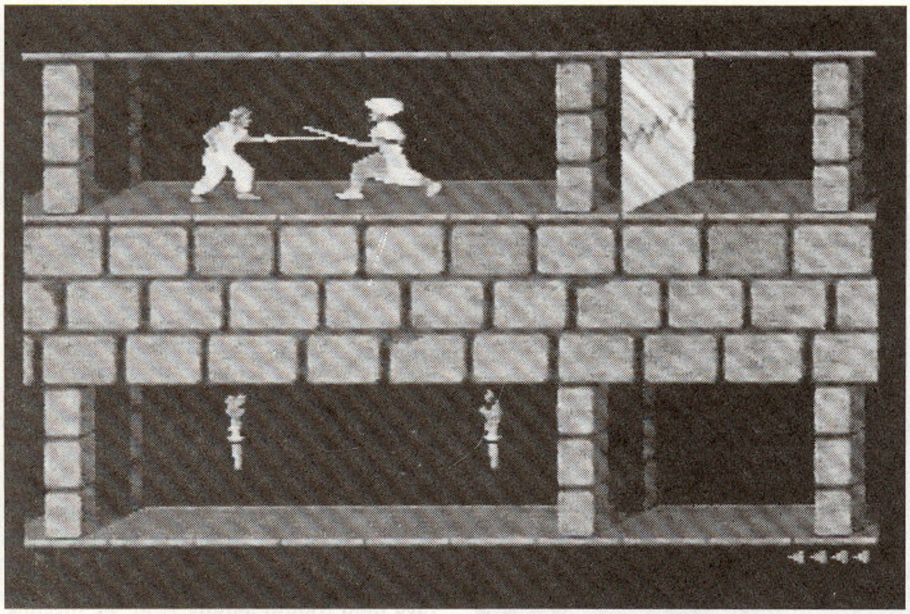
1/1. Ez egy plusz szint. Itt csak üvegek vannak, de nem mindegyiket szabad meginni. A szint elején kiíródik az a betű, amit valamelyik korsón láthatunk. Ez lehet C, G, A stb. Az aktuális betűvel jelzett korsót szabad felvennünk. Az ital elfogyasztása közben az ajtó kinyílik.

2. Induljunk el balra, és kardozzunk a katonával. Fussunk balra a karókig. Lépkedjünk át rajtuk, és szaladjunk ismét balra. Másszunk fel a felettünk levő szintre (emeletre), és gombbal ugorjunk át a másik oldalra. Másszunk fel két emeletet, és fussunk balra. Az őr megsemmisítése után nyissuk ki a rácsot és másszunk fel. Szaladjunk balra, nyissuk ki a rácsot, és fussunk tovább. Kardozzunk a katonával, menjünk balra (az üvegeket ne szedjük fel) és másszunk fel. Lépkedjünk balra, és helyből ugorjunk át a szakadékon, öljük meg az őrt. Fussunk balra.

A felső emeleten találhatunk egy korsót, ami extra energiát tartalmaz. Ezt nem szükséges felvenni ha trainer verzióval játszunk. Szaladjunk balra, addig amíg egy szakadékhoz nem jutunk. (Vigyázzunk a karókra!) A szakadékot helyből ugorjuk át. Másszunk fel két emeletet, és az előbbi módszerrel ugorjunk vissza. Fussunk jobbra, másszunk fel, és kardozzunk az őrrel. Lépjünk az ajtónyitóra, és ugorjunk vissza. Szaladjunk az ajtóig jobbra.

3. Először jobbra fussunk, és másszunk fel két emeletet. Ugorjunk át. Másszunk fel három szintet és ugorjunk át a szakadékokon, majd lépjünk a nyitóra. Ezt ismételjük meg, gyorsan ugorjunk át a szakadékokon. Szaladjunk balra, nekifutásból ugorjuk át a szakadékot, és fussunk balra, majd az előző ugrást ismételjük meg (gombbal). Nehezen tudunk csak megkapaszkodni a rács alatti kőlapon. Másszunk fel, és szaladjunk balra. Másszunk fel a szemközti rácshoz és nyissuk ki. Menjünk balra, és óvatosan lépjünk át a vasfogon, majd ereszkedjünk le három emeletet. Fussunk jobbra és lépjünk a kőlapra, ami az alattunk levő ajtót nyitja ki. Másszunk fel három emeletet, és menjünk jobbra addig, ameddig egy csontváz-





zal nem találkozunk. A kardunkkal lökjük le a szakadékba a csontvázat, majd másszunk le. Ismét lökjük le. Szaladjunk balra az ajtóig.

4. A palotában vagyunk. A jobb oldali nyitóra lépjünk rá, és fussunk balra. Ereszkedjünk le két emeletet, és menjünk balra. Kardozzunk a katonával, és ugorjunk át a szakadékon, majd ismét fussunk balra. Másszunk fel a felső szintre, a másik oldalon le, és lépjünk a rácsnyitóra, majd az előzőket ismételjük meg. Szaladjunk vissza addig, ahol az őrrel harcoltunk, és ugorjuk át a szakadékot. Menjünk jobbra és ugorjunk át a felhúzott rács alatt, majd szaladjunk jobbra és ismét ugorjunk egyet. Másszunk le, majd fel. Óvatosan lépjünk át a vasfogon, és küzdjünk meg az őrrel. Másszunk jobbra, kardozzunk az őrrel, majd ismét menjünk jobbra. Másszunk fel és fussunk a szakadékig, majd ereszkedjünk le egy emeletet.

Lépjünk át a vasfogon, nyissuk ki az ajtót, másszunk vissza. Szaladjunk balra és ugorjunk át nekifutásból a tükrön. Itt láthatjuk a tükörképünket, aki még sok bajt fog nekünk okozni. Szaladjunk vissza addig, ahol az ajtó található.

5. Jobbra, fel. Kardozzunk az őrrel, lépjünk jobbra, és másszunk fel két emeletet, majd még egyet. Ha jobbra megyünk, a felső szinten egy korsót látunk. Ezt sajnos nem tudjuk megszerezni, mivel a tükörképünk elviszi előlünk. Tehát ne jobbra, hanem balra menjünk. Nekifutásból ugorjunk át a másik oldalra, ereszkedjünk le. Menjünk balra és öljük meg az őrt. Ereszkedjünk le és nyissuk ki a rácsot. Szaladjunk balra, és küzdjünk meg az őrrel. Ugorjunk át a szakadékon. Innen már egyszerűen el tudunk jutni az ajtó kinyitójáig.

6. Lépkedjünk balra (nyitó, karók), ugorjuk át a szakadékot. Menjünk balra és küzdjünk meg az őrrel és verjük le a szakadék feletti kőlapot. Fussunk balra, ugorjunk. Tükörképünk egy hatalmas szakadék mellett áll. Helyből ugorjunk (gombbal), egy félelmetes zuhanás után, éppen hogy megtudunk kapaszkodni.

7. Ismét a börtönben vagyunk. Másszunk fel, forduljunk meg és ugorjunk át (gombbal) a másik oldalra. Szaladjunk jobbra, küzdjünk meg az őrrel, lépjünk át a vasfogon. Fussunk jobbra és nekifutásból ugorjunk át a kőlappal fedett szakadékot, majd lépkedjünk jobbra. Ugorjunk fel, és mozgassuk meg a plafont fedő kőlapot. Másszunk fel a keletkezett nyíláson. Fussunk jobbra (vasfog, katona), majd ereszkedjünk le egy szintet. Ismét haladjunk jobbra. Lépjünk a rácsnyitóra. Menjünk át a rács alatt, lépjünk a második rács nyitójára, majd a következő kőlapra. Helyből ugorjunk (karók, rácszárók). Szaladjunk jobbra, lépjünk a három nyitóra, és ereszkedjünk le (gombbal kapaszkodunk) két emeletet.

Menjünk balra. Ugorjunk át két kisebb tüskés szakadékot, lépkedjünk át a karókon. Másszunk fel három szintet, szaladjunk jobbra, felmászás után küzdjünk meg az őrrel. Ereszkedjünk le két emeletet (vasfogak), majd fussunk balra. Nyissuk ki az első rácsot, lépjünk a második kőlapra, és helyből ugorjuk át a rácszárót. Szaladjunk balra. A szakadék előtti nagy korsót vegyük fel, és igyuk meg a benne levő italt. A felcsendülő zene közben ugorjunk le a mélységbe. Nyissuk ki a rácsot, majd az ajtót.

8. Másszunk fel, menjünk balra (őr), és helyből ugorjunk át (gombbal). Ereszkedjünk le, ugrás le. Szaladjunk jobbra (vasfog, karók, katona). Nekifutásból (gombbal) ugorjuk át a szakadékot, másszunk fel két emeletet, és helyből (gombbal) ugorjuk át a szakadékot. Ha továbbmegyünk, balra guggolgatással haladjunk, mert különben az őr megölhet. (Guggolgatás = a botkormány le+sré irányú mozgatása a tűzgomb folyamatos nyomogatása közben.)

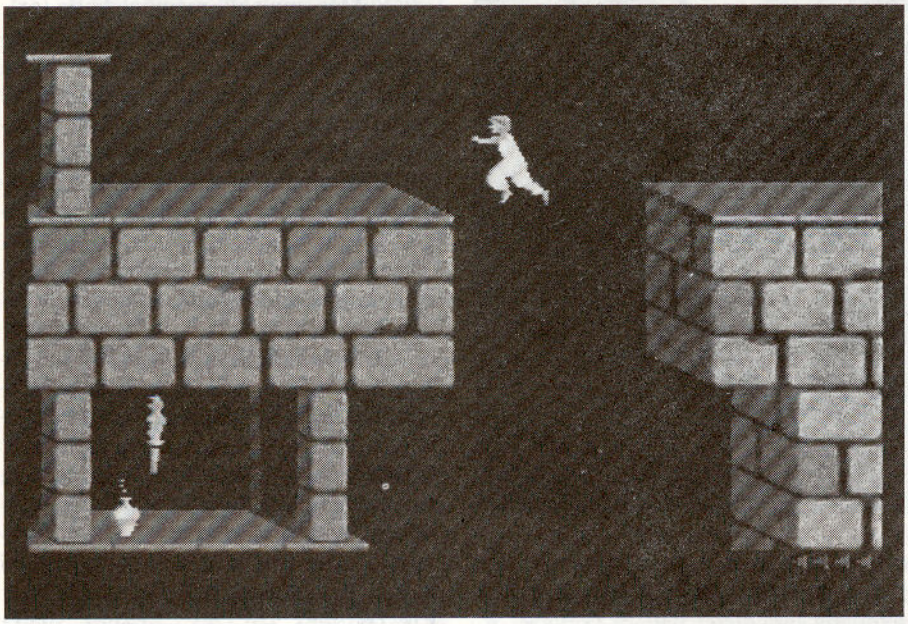
A következőket gyorsan kell végrehajtanunk: nyitó, ugrás (gombbal), vasfogak, nekifutásból átugrás (kétszer). Haladjunk balra (nyitó, vasfog, őr, záró). Ugorjuk át (gombbal) és lépkedjünk át a vasrácsokon, nyissuk ki az ajtót. Ha viszszamegyünk, a rácsot zárva találjuk. Várjunk! Hamarosan megérkezik a hercegnő kisegere és rászalad az ajtónyitó kőlapra. Ereszkedjünk le, és fussunk balra (karók).

9. Menjünk balra (vasfog, rács) és másszunk fel két szintet. Forduljunk meg és ugorjuk át a szakadékot. Szaladjunk jobbra (vasfog, katona). Másszunk fel öt emeletet és ugorjuk át a kőlapokkal fedett szakadékot (őr, ajtónyitó). Hasonlóképpen menjünk vissza, ereszkedjünk le két szintet, szaladjunk balra. Az oszlopok mellett a kőlap megmozdul és ráesik egy nyitóra. Ugorjunk és fussunk jobbra.

Másszunk fel öt emeletet, ugorjunk át és nyissuk ki a rácsot, majd haladjunk balra (katona). Ugorjunk át, és másszunk fel két szintet, és nekifutásból ismét ugorjunk át (gombbal) a rácsnyitó kőlaphoz. Ereszkedjünk le, és szaladjunk balra. Ha átugorjuk a vermet és a felső úton megyünk tovább, három korsót találunk, de csak a középsőt szabad meginni.

Tehát ereszkedjünk le öt szintet. Lépkedjünk át a vasfogon, nyissuk ki a szemközti rácsot, a felső rácsot húzzuk fel. Másszunk fel az említett rácshoz, fussunk jobbra (vasfog, katona). Lépjünk az ajtónyitóra és ereszkedjünk le.

10. Jobbra induljunk (őr, nyitó). Ne másszunk fel, hanem fussunk jobbra. Most másszunk fel egy szintet, és lépkedjünk át a vasfogakon. A második vasfog előtti kőlapról ugorjuk át a rácsnyitót. Menjünk jobbra fel. A vasfog előtt mozgassuk meg a mennyezetet, és másszunk fel a keletkezett nyíláson. Nagyon gyorsan szaladjunk balra. A leeső kőlappal mi is lezuhanunk (őr). Ezen a szinten men-



jünk balra (nyitó, záró). Fel (katona), majd a folyosó végén mozgassuk meg a plafont. Ha felmászunk a keletkezett nyíláson, megtaláljuk a rácsnyitót. Ereszkedjünk le és szaladjunk balra az út végéig. Nyissuk ki az ajtót (katona).

11. Fel, majd jobbra. Nekifutásból ugorjunk át (gombbal). Az oszlopoknál a mennyezetet mozgassuk meg, és mászszunk fel a keletkezett nyíláson. Menjünk jobbra, ha leestünk vívjunk meg az őrrel (vasfog). Fussunk jobbra. Az oszlopoknál a plafont meg tudjuk mozgatni. Másszunk fel a nyíláson és szaladjunk jobbra. A leeső törmelékek hatástalanítják az ajtózárókat. Fussunk balra (ugrás, nyitó), majd fordoljunk vissza és menjünk jobbra (leeső kövek, lecsukodó rács). Másszunk fel és kardozzunk az őrrel. Nekifutásból ugorjunk egy hatalmasat. A padló labilis, csak nagy nehezen tudunk megkapaszkodni. Fel (katona), le (nyitó). Fussunk vissza a kijárathoz.

12. Induljunk el jobbra, menjünk át az alagút alatt. A következő toronyszerű képződményen másszunk fel két emeletet, lépjünk hármat jobbra, és forduljunk meg. Másszunk fel három emeletet, majd a tüskéken áthaladva nekifutásból ugorjuk át a hatalmas szakadékot. Célszerű a tűzgombot használni.

Másszunk fel három emeletet, és nekifutásból ugorjunk át az első, a második, majd a harmadik toronyrészletre. A torony belső járatában vagyunk (egy csontvázat, és két égő fáklyát látunk). Másszunk fel egy szintet és mozgassuk meg a plafont. A keletkezett nyíláson másszunk fel még két szintet. Lépkedjünk balra, és helyből ugorjuk át a szakadékot. A tűzgomb használata itt is lényeges! Másszunk fel egy emeletet és gyorsan fussunk balra. A két torony közötti szakadékot itt is, ugyancsak nekifutásból tudjuk átugrani. Másszunk fel négy emeletet, és lépjünk rá a mellettünk levő rácsnyitóra.

Ereszkedjünk le egy szintet. Lépjünk balra négyet, forduljunk meg és nekifutásból ugorjuk át a szakadékot. Szaladjunk balra a másik toronyig. Másszunk fel egy szintet. Mozgassuk meg a faltól számítva az ötödik kőlapot. Másszunk fel a negyedik kőlapra és nekifutásból ugorjunk át a másik oldalra. Fussunk balra. Másszunk fel egy szintet. Menjünk balra (két kőlap leesik), ekkor megjelenik a tükörképünk. A kardot a gép automatikusan elővesz (így a hasonmásunk is), ha harcolunk magunkkal, hamarosan meghalunk.

A következőket tegyük: a kard előhúzása után rögtön tegyük is el, és fussunk balra a tükörkép felé. Érdekes dolog történik, aminek köszönhetően nekifutásból beleugorhatunk a szakadékba. Egy kőlapra ugrottunk, és evvel a játék egy plusz szakaszába értünk. Ez egy nagyon könnyű rész, lezuhanó kőlapokat kell kikerülnünk, de itt vár a fő ellenség, akiről bővebben a játék kedvéért nem kívánunk szólni. A legutolsó szint már nem tartozik az előzőhöz, ez az előbbinél egy kicsivel egyszerűbb. Sok sikert kívánunk.

GLOBÁLIS GLADIÁTOROK

McDonald's óriási heccre invitálja a nagyérdemű gyereksereget. Fődíjként 2 ,,Royal TS" és egy 9-es csomagolású csirke integet! Kechup-pal és Mayonnaise szósszal felfegyverkezve. Egy pillanatra sem habozik a két egycsapásra világhírűvé vált McDonald's hős Mick és Mack hogy ezen a versenyen részt vegyenek. Egy mintaszerű ,,mászás-futás" során egyik a két tornacipős hős közül rágógumit rágva több plattform által díszített jeleneten keresztül keresi a kis McDonald's jelképet M-t.

Az utazás színes világokon keresztül zajlik, mint a ,,rossz-világ" egy mágikus erdő vagy toxi-város, egy riasztó nagyváros. A plattformok természetesen a környezethez illenek. Így fut a játékos zöld mezőkön, sötét poklokon át vagy szökdécsel a fákra, ill. az építkezési állványok között. Csak mikor Mick és Mack meghatározott számú, bonyolultan elhelyezett és nehezen elérhető szimbólumot összegyűjtött, akkor nyomulhat a szint végéhez — és egyenesen személyesen Ronald McDonald karjaiba, aki a célzászlót lengeti nekik.

Buta módon néhány különösen undorító teremtmény életcéljának tekinti-e gyűjtési akció megakadályozását. Legyenek bár csapkodó Ichtyosauriusok, szemeteskukák vagy húsevő növények egy mindenképpen közös bennük: mindegyik mindenféle veszélyes tárggyal dobálódzik. Gyakran a muníció zöld és nyákos és egészen undorító foltokat csinál és Oriel Ultrával sem távolítható el. Mivel Mick és Mack ezt tudja és csontjukig hatol a félelem így minden egyes találkozással rengeteg energiát veszítenek életerejükből. Piros szívek gyűjtésekor felfrissülnek ugyan, de jobb félni, mint megijedni. Gyakran ugyan elég, ha az ember a dühös jószágot megérinti, mert ezek egy tiszta



ifjúval történő bőrkontaktus (érintkezés) kapcsán jelentősen veszítenek életkedvükből.

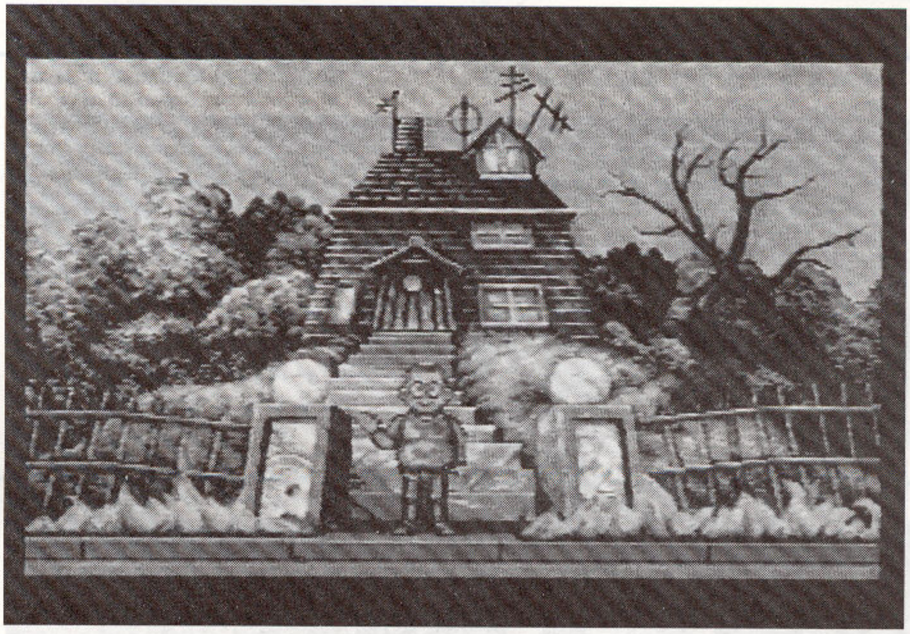
Egyébként időről időre feltűnnek a játék Sega-Versiói (Mega Drive, Master Systems, Game Gear) ez nem csoda, mivel az Amiga-Versióban terjedelmes beállítási menüt integráltak. Maga a zene és hangmenü sem hiányozhat. Itt minden hallható, ami a játék során adatbevitelre kerül.

Úgy 3000-nél már bizonyos bukásokról lemondhatunk (?), amely egyébként az AMIGA 1200 esetében érvényes! Akkor sem, ha a játék 10 perc alatt töltődik.

MORPH

Morpheus a görög álomisten és az alvásisten fiy Hypnos a XX. században sokféle felfogást élt át. Így történt, hogy az Ópium fő alkotóelemét Morfiumra keresztelték, mivel ez általában fájdalomcsillapító hatású és erős víziókat okoz.

A morfológia, mint névrokon a dolgok alakjával és viselkedésével foglalkozik. Így alakult ki az első komputerjáték, mely a Morpheus nevet viseli.

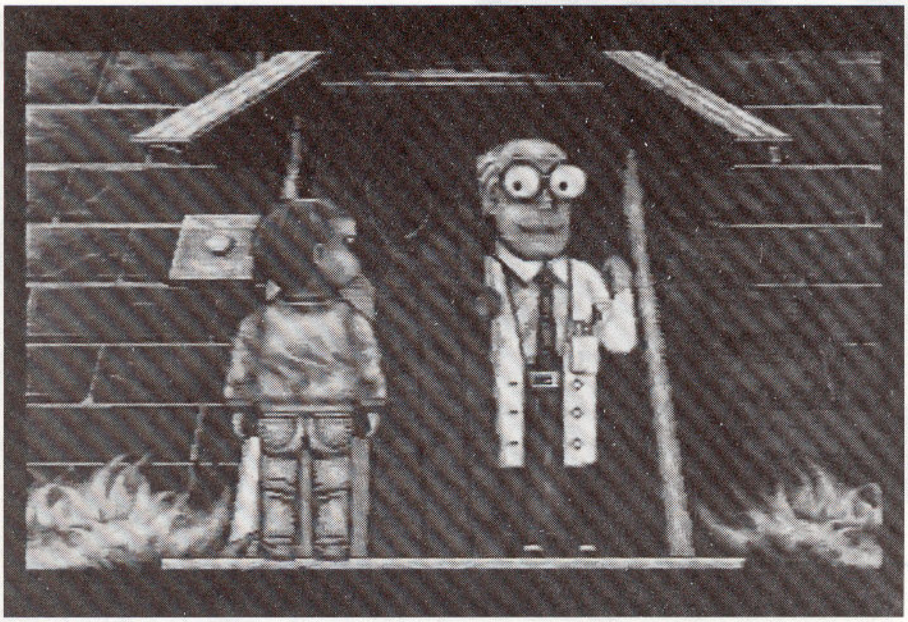




Peter Johnson a kezdeti idők Arkanoid-, Wzball-, és Robocop forgalmazója a Milleneumkor a Morph-val debütált.

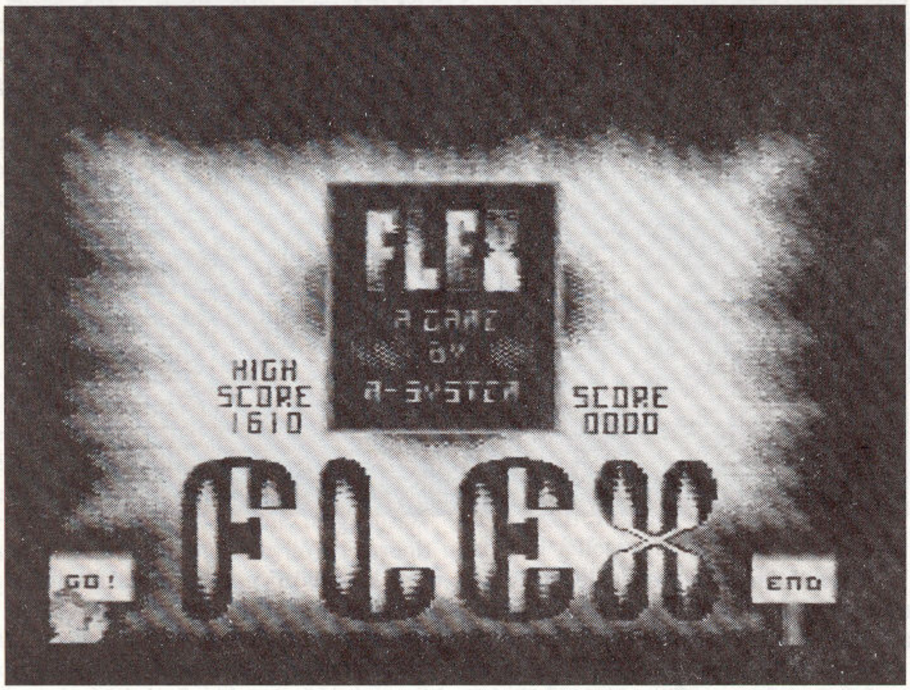
Parányi Bob nagybácsikájával tengeti napjait és nagyon segít a zseni Hirnischmalznak Teleport gépe elkészítésében. Mikor a gép végre elkészül és működik, Bob is ki akarja próbálni és kegyetlen vége lesz: Molekulái tervszerűen szétesnek és már éppen raknák össze, amikor egy villám belevág a gépbe és az darabjaira esik szét. A sajátos köztes molekulákban dimenzók vannak, Bob kénytelen összegyűjteni a gép részeit és e célból négy különböző golyóvá átalakítani, ill. morfálni: Víz-, gumi-, felhő- és Bowlinggolyóvá.

24 szinten kell Bobnak a golyókat a bizonyos idő alatt legyőzni és csak egy szinten belül korlátozva gyakran változtatni. Sebességtesztek mellett a játékos helyenként taktikai meggondolásokat alkalmaz. Útközben összetalálkozik (ütközik) kártevő idegen műszerekkel, mint pl. tüskékkel, ventilátorokkal, melyek a tulajdon halálukat a golyók mindenkori állapotától teszik függővé. Mindazonáltal nemcsak mogorva készülékekkel kerül



érintkezésbe, hanem segítőkészekkel is: kis extrák mellett kiegészítő átalakulások sodorják Bobot, légfúvókák és csövek segítik továbbjutását eldugott Bonus-Terekbe. A legfontosabb a tanszformátorok, melyek őket automatikusan egy másik elemi golyócskává változtatják. Csak ügyes trafók használatával lehet a legtöbb szintet feloldani. Akinek ez nem sikerül az örül a véget nem érő folytatásnak.

FLEX



Bizonyára mindenki találkozott már nagyapáink egyik kedvenc játékával, a Tili-tolival. Aki nem ismerné erről a névről, annak is biztos beugrik a következő pár sor átfutásakor. Általában 4×4 kis lapocskából áll — ez lehet fa, vagy műanyag — és egy keretből, ami összefogja, hogy ne essen szét és megakadályozza az esetleges türelmetlenségből adódó visszaéléseket.

A lényege, hogy a lapocskákra rajzolt ábrát, számsort, esetleg szöveget jól összekeverjük és megpróbáljuk kirakni egy hiányzó lapocska segítségével.

Ezt a játékot már kipróbálhattuk a TV2-n, de aki nem szereti a nyilvánosságot, vagy nem sikerült telefonálnia, az most kipróbálhatja C64-en is.

A játék egyszerű, 15 db számot kell a megfelelő sorrendben kirakni, miután a gép kellőképpen összekeverte azokat. Túl sokat nem gatyázhatunk, mert a képernyő alján mászkáló kedves jószág ugyan csiga, de biztosan futóedzéseket vett, mert szinte robog a cél felé, ami neki pihenőt, nekünk viszont a játék végét jelenti. Ha sikerül idő előtt kiraknunk a számsort, atlétánk ugrálásával fejezi ki örömét, hogy nem kell tovább ,,futnia''. A kényelmesebbek választhatnak örökidőt, így csigánk ,,helybenfut''.

Az egyes pályák a számsorok variációival tarkul, ami egy idő után kifakul, sőt monoton lesz. Ha a siker után kapnánk valamilyen szép képet, akkor megéri a fáradozást, de így egy cseppet unalmassá vált.

Viszont a jó zene, ami nem őrjítő húsz perc múlva sem, a borzasztóan klassz háttér és a kitűnő játszhatóság kárpótól egy ideig ezért a kis csekélységért.

A programot az A-SYSTEM írta.

BBEE



KLOSS KAPITÁNY

Ebben a kitűnő játékban melyet az LK Avalon csapat írt, a II. vilgháború idejére képzeljük magunkat. Az akkori Németországban különböző titkosfegyverek tervezése folyik. Hősünk azt a feladatot kapta, hogy ellopjon egy ilyen tervet a hozzátartozó műszaki leírással együtt. Ezt egy többszintes földalatti betonbunkerben kell 10 darabból összeszedni a dokumentummal együtt és a bunkeren adott helyre vinni. Hogy ez ne legyen ilyen egyszerű arról a programozók messzemenőkig gondoskodtak. Elhalálozásunk akár egyetlen hiba miatt is bekövetkezhet, ami 9 tervdarabkánál már-már a mosdatlan szavak használatát teszi kilátásba.

Halálesetek:

Elfogy az ennivaló. Ezt egy csirkecomb szimbolizálja. Elfogy az innivaló. Ez egy csésze kávé alatti számból derül ki (a kajánál is). Kilencvenkilencről nullára redukálódik mindkét érték. Mivel drinkbárt és gyorsbüfét nem találtam sehol, kénytelen voltam beérni a termekben megbúvó combokkal, kávékkal. Ezeket nekünk hagyhatták itt mert az egész játékban egyetlen egy élő emberrel, de még halottal sem találkoztam, csak húsz cm magas forgó kis bosszúságokkal, amivel ha testi kontaktust létesítünk energiánk rohamosan fogyni kezd. Tehát az ennivaló rejtélye megoldatlan. Az meg abszolút nonszensz, hogy az ajtókat kulcsrazárják ugyan, de gondoskodnak a kulcsokról, nehogy ne tudjunk valahova bemenni. Ezt is nyilvántartja a program (kulcs ikon).

Egy apró kellemetlenség is társult az ingyenkonyhához. Néhány helyen sétálgatásunk egy sortűzzel ér véget. Persze kivédhetjük az ólommérgezést, ha az érzékelőket egy ideig kikapcsoljuk a falon levő szerkezettel.

Viszont jó hír, hogy emberünk akármilyen magasról leugorva sem hal meg.

A játék a következő billentyűket használja a Joy-on kívül:

F1: abort
F3: zene/hanghatás
F5: az eddig összegyűjtött dokumentumokat nézhetjük meg
F7: pause. Hősünk egy cigarettára gyújt.

A játék használhatósága tökéletes, grafikája nagyon jó, a zenét meg mindenki kikapcsolhatja, ha már unja. Az animáció iszonyatosan jó. A dohányzás, a füst, a mozgás, sőt a hányás ha emberünkkel több csirkét eszünk a kelleténél, egyszerűen marha jó.

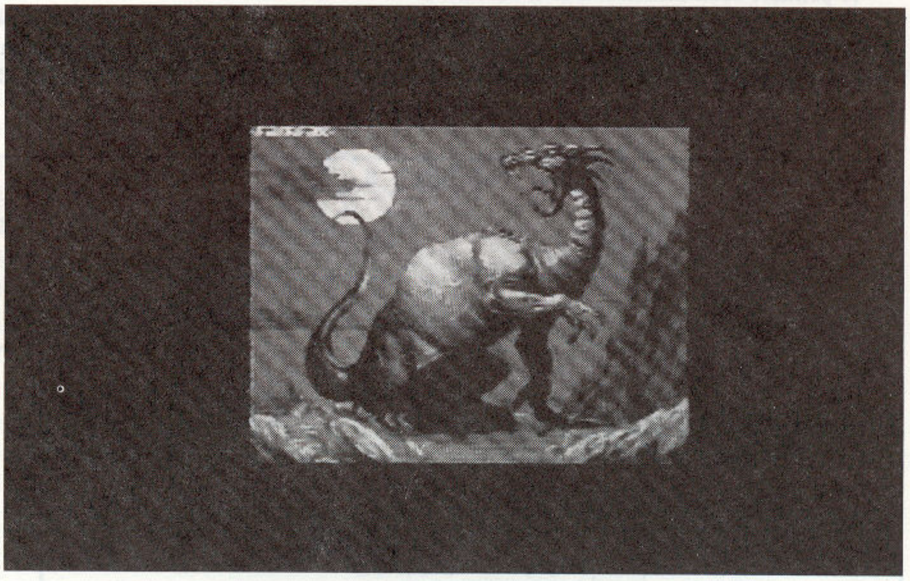
Egyedüli hiányosság, hogy a játékos semmi tájékoztatást nem kap feladatáról, így a hangulat kicsit hűvös.

Egyébként egy nagyszerű mászkálós program lövöldözés, emberirtás, erőszak nélkül.

BBEE

MIRROR

ANDROMEDA produkció. Ez egy igazi norvég zenelemez. Igen apró részletekig kidolgozott. A zenék, képek, srollok külön menüben állítgathatók. A vektorgrafikai megoldások Mr. Hyde, a gyönyörű picture-ök Fairfax nevéhez fűződnek. A demó kilenc dallamos zenét tartalmaz.



Az MPS 1230-as nyomtató módosítása

Az MPS 1230-as típusú mátrixnyomtató a COMMODORE cég egyik legjobban sikerült gyártmánya.

Nemcsak a C-64-hez használható, hanem az egyéb számítógépekhez is!

A gyakori használat során azonban néhány apróbb problémát fedeztem fel.

Az alapbeállításban (értsd: szállítási helyzet) a fej tűinek ütéserőssége nagyon nagy. Előnye, hogy az ún. Sirály papírból jó (pl.: Pelikán Ultrafilm 10) indigóval akár három (!) példányt is átvisz gond nélkül (miközben a precíz mechanika jóvoltából a lapok nem csúsznak el). Ennek az a hátránya, hogy gyakori (napi egy-másfél órai) használat után a kazettában lévő szalagot, annak összehegesztése mentén, kilyukasztja. Nagyon kellemetlen, miközben — mondjuk — levelet írunk.

A fenti hibát szervizben tudják elhárítani, de sokpéldányos másolás esetén nem érdemes.

Szerezzünk be inkább tartalékba: GR.673 típusú festékkazettát. Ennek ismert változata az OLIVETTI DM100 GR.673. A célnak bőven megfelel, és olcsóbb mint az eredeti COMMODORE!

A kazetta házi töltését NEM ajánlom; mert, ha rosszul rakjuk vissza a szalagot: 1. megszorul a mechanika, 2. elszakad a szalag!

A másik probléma, amely a másfél éves — fokozott — használat során felmerült:

A kijelzőegység alatti ,,billentyűk'': fóliaalapú membránbillentyűk.

Előnye: olcsó.

Hátránya: kilyukad.

Nálam is erre a sorsra jutott a gép (természetesen egy fontos levél közepén, este 10-kor). A következőképpen lehet ezt elhárítani:

Anyagszükséglet:	villogó LED (piros)	2 db
	zöld LED	1 db
	mikrokapcsoló	3 db
	szalagkábel (10 erű)	0,2 m.

A printert csatlakoztassuk le a hálózatról és a C-64-ről.

Vegyük le a traktort a lapvezetővel együtt.

Húzzuk le a lapbefűző gombot.

Fordítsuk fel, majd a — most már — jobb oldalt található fülecseket egy kisméretű csavarhúzóval pattintsuk ki. Ugyanezt a baloldalon megismételve, a printer világos teteje könnyen levehető. (Ne erőltessük, nehogy a laprögzítő kar eltörjön!)

Az alaplap bal felső sarkán lévő szalag-NYÁK-kábelt finoman felfelé húzzuk le. Ezután a gép két részben van, a tetején a kijelző+vezérlő panel, aljában a többi.

A kijelzőpanelt pattintsuk ki.

Plexiből vágjunk egy ugyanolyan méretű darabot. Ebbe könnyedén ,,bepakolhatjuk'' az ábra szerinti sorrendben az alkatrészeket.

Fentről lefelé:

LOCAL: villogó piros LED, azt jelzi, ha a C-64 még nincs bekapcsolva.

ON: zöld LED: a printer bekapcsolt állapotát jelzi.

FAILURE: villogó piros LED, a papír helyzetét jelzi (nincs papír, v. a BE (= LOCAL) gomb nincs benyomva).

LOCAL: mikrokapcsoló, bekapcsol, ha van papír.

LINE FEED: mikrokapcsoló, 1 sort emel.

FORM FEED: mikrokapcsoló, teljes lapot dob.

(A fenti LED-eket a bordázott részre, a kapcsolókat a bal alsó sarokba tettem, így könnyebben kezelem.)

A vezeték bekötését a rajz alapján hajtsuk végre, majd egy próbaüzem következzen.

Ha minden jó = a tetőrész visszapattintható. (Javaslom a feliratozást újra megoldani!)

A hibátlan üzem esetén:

— bekapcsolás után a két piros LED villog, a zöld ég,
— a C-64 bekapcsolása után a felső kialszik, csak az alsó villog (a zöld jelez),
— a papír befűzése (és a LOCAL = BE kapcsoló megnyomása) után a piros LED-ek nem villognak,
— ha a papír kifogyott, az alsó piros LED villogása jelzi azt.

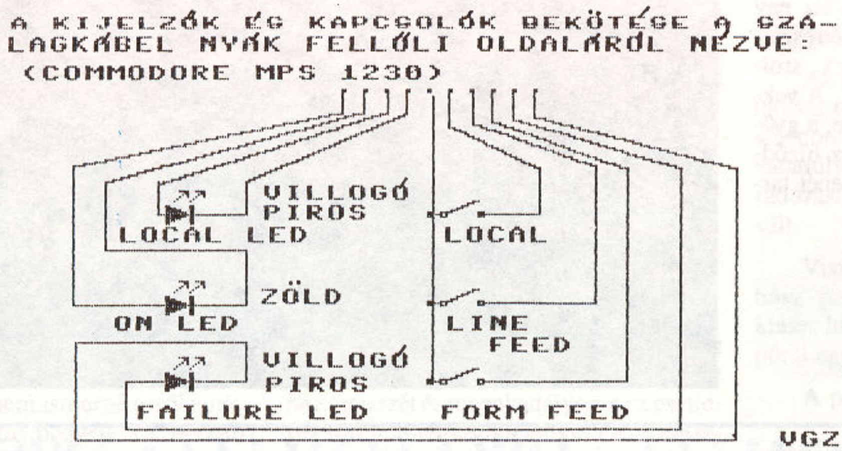
A printer kezeléséhez néhány tanács:

— havonta porszívozzuk ki,
— másolópapírt használjunk (ez nem porzik),
— jó minőségű indigóval (lásd fent) dolgozzunk,
— tartalékkazetta (töltve is megfelel) legyen kéznél,
— a szereléseket ÁRAMTALANÍTOTT gépen, KISTELJESÍTMÉNYŰ pákával végezzük.

A printer a fentieket meg fogja hálálni!

Jó munkát kívánok:

Varsányi Gábor



Az MPS 1230-as nyomtató új kezelőszervei és kijelzői

Commodore újságok 1986—90-ig eladók 30 Ft/példányonkénti áron, együtt 20 Ft/db. Nagy Péter, 1056 Budapest, Molnár u. 20.

C-64-esek figyelem!! Vadonatúj 2 oldalas lemezek 50—90 Ft közötti áron eladók (pl.: NONAME, BASF, 3M stb.). Minél hamarabb írj, mert korlátozott számban van! A borítékra írd rá „lemezek". Papp Mihály, 4034 Debrecen, Kerecsen u. 37.

Keresem C-64-re a CATALYPSE, az R-TYPE és a STRATOFIGTHER című programok magnós változatát megvételre vagy cserébe. Szilágyi Róbert, 3300 Eger, Vers Péter u. 9. fszt. 3.

C-64 és C+4 kazetták 200 Ft/db, lemezek 50 Ft/db áron! Listát küldök. Csernáth Zoltán, 1171 Budapest, Dalnok u. 5.

C-64 kazettán keresem F-14 TOMCAT. Ugyanitt programok eladók. Válaszboríték. Farkas Dániel, 7400 Komló, Vértanú u. 5.

C-64-re már 3 Ft/db ártól eladók programok lemezre és kazettára nagy választékban. Válaszborítékért listát küldök. Kovács Krisztián, 6750 Szeged, Bartók B. u. 6.

PLUS/4 és C64-es gépekre program eladás, és csere! Programkészlet, 2—2000 prg. DEMO-t nem gyűjtök! Válaszboríték esetén részletes listát küldök. Novák Péter, 1132 Budapest, Kresz Géza u. 21.

Kedves olvasónk! A C-64-AMIGA rovatban megjelent és egyéb játék-, felhasználói, valamint demo programok megrendelhetők. Hetente bővülő választék, precíz (vízhatlan) csomagolás, kedvezmények (AMIGA). Bélyeggel ellátott válaszborítékért részletes tájékoztatót és programlistát küldünk. Kérjük tüntesse fel a számítógép perifériáit (kiegészítőit), típusát. Amigán a lista mágneslemezen is kérhető egy 3.5-ös lemez, és 40 Ft-os bélyeg ellenében. C-64-en vállalunk epromégetést, cartridge készítést, valamint (adatbáziskezelő, nyilvántartó) készítését. A borítékokat géptípusonként a következő címekre lehet feladni:
AMIGA: FRIEND TWO CREW 1399 Bp., Pf.: 701/55 Tel.: 1-351-289 (naponta 16—20 óráig)
C-64: BUMBLE BEE 1116 BP., Sáfrány u. 44. Tel.: 1-627-724
Várjuk olvasóink észrevételeit postacímünkre (1399 Bp., Pf.: 701/55).

C64-es programleírások 40 FT-ért! Válaszborítékért lista! COMPI—CREW 9730 Kőszeg, Deák F. u. 3/E.

576KByte-ban megjelent és egyéb C-64-es programok eladása lemezen és kazettán. Lemezes programok kazettás változata is! Érdemes! Deutsch Szabolcs, 8749 Zalakaros, Fő utca 19. Tel.: (93) 340-481

Elektronikai-nyilvántartó programok, C-64-re. Válaszborítékot kérek. Varsányi, 9700 Szombathely, Nagy L. u. 11.

C-64-re programok 15 Ft-ért eladók! Válaszborítékért lista! COMPI-CREW, 9730 Kőszeg, Deák F. u. 3/E.

C-64-re programok készítését vállaljuk! Válaszborítékért árlista. Varga Soft, 9730 Kőszeg, Petőfi tér 26.

C64-re leírások, oktató-, felhasználói- és játékprogramok 5 Ft-tól. Borítékért lista. Földesné, Szolnok, Győrffy u. 12. I./7. Tel.: 56/420-544.

C-64 + floppyra eladó angolt tanító szókimondó program. Ára: 390 Ft + portó. Sándor Lajos, 5836 Dombegyház, Ifj. lt. A/3.

Ha kivágta
a megrendelőlapot,
kérjük tegye
felbélyegzett borítékba,
címezze meg,
és úgy küldje el.

Köszönjük!

IGEEEN! Megrendelem az

ACOMP Kft.

borító hátoldalon szereplő kínálatából
a feladó címére postai utánvétellel az alábbi termékeket:

.... db .... Ára .... Ft/db: összesen .... Ft
.... db .... Ára .... Ft/db: összesen .... Ft
.... db .... Ára .... Ft/db: összesen .... Ft
.... db .... Ára .... Ft/db: összesen .... Ft
.... db .... Ára .... Ft/db: összesen .... Ft
.... db .... Ára .... Ft/db: összesen .... Ft
.... Mindösszesen: .... Ft

Tudomásul veszem, hogy a csomagküldés átfutási ideje 2-4 hét.
50.000,- Ft feletti megrendelésre csomagküldést nem vállalunk!
Az áraink a postaköltséget nem tartalmazzák!

Tippek–trükkök

Gyorsabb programok

A gyakrabban használt számokat, címeket és változókat a programok elején kell definiálni. Különösen akkor, ha azok ciklusokban is előfordulnak. Így a ciklusok sokkal gyorsabban futnak le. Például:

```
10 A = 100
20 FOR I = 1 TO 5234
30 A = A + 1.3456 * I)
40 B = PEEK (A + 12 * I)
50 NEXT
```

Ez a program sokkal lassabban fut le, mint az alábbi változata:

```
10 A = 100: B = 0: C = 1.3456: D = 5234: F = 12
20 FOR I = 1 TO D
30 A = A + C * I
40 B = PEEK (A + F * I)
50 NEXT
```

Gyorsabb programok, 2. rész

A szorzásoknál is jelentős futási idő takarítható meg a C64 olyannyira értékes gépidejéből. Célszerű, ha a több számjegyű szám a szorzandó, ennek megfelelően a kevesebb számjegyű szorzó áll hátul. Vagyis a

6.15442324 * 4.65

gyorsabban számítódik ki, mint a

4.65 * 6.15442324

Hasonlóképpen nagyon meggyorsítja a program futását, ha a hatványozást többszörös szorzásként végeztetjük el. Tehát nem

X = 2↑4

hanem

X = 2 * 2 * 2 * 2

Ez utóbbi módszer természetesen csak egész értékű hatványra emeléskor használható.

Programok a RAM-kiterjesztésben

C128 módban a Basic programok átmenetileg a 17XX címen kezdődő RAM-kiterjesztésben tárolhatók. Általában erre a célra szolgál az alábbi utasítás sorozat:

```
X = PEEK (4624) + 256*PEEK(4625) — (PEEK(45) — PEKK (46)*256)
STASH X, 7169,0,0
PRINT X
```

Az X változó értékét meg kellene jegyezni, mivel ez a szám a program visszahozásához ismét szükséges lesz. A

```
FETCH X-érték, 7169,0,0
```

utasítással lehet a programot ismét az operatív memóriába visszatölteni. Ez azonban csak akkor funkcionál, ha a Basic program nem hosszabb 46079 bájtnál, illetve, ha grafikát is tartalmaz, akkor 36863 bájtnál.

Ha azonban a program ennél hosszabb, akkor az operációs rendszer hibája miatt az 53248 memóriahelytől kezdve már nem a Basic-RAM helyezkedik el, hanem az I/O-terület. Megkísérelve az ilyen hosszú program visszatöltését, a gép lefagy, mivel az I/O rekesz átíródott.

A következő asszembler-program megszünteti ezt a hibát:

```
,,003f0 sei          ; megszakítás lezárva
,,003f1 ldx $ff00    ; memória konfiguráció megjegyzés
,,003f4 sty $df01    ; dma-vezérlő beírása
,,003f7 ora # $01    ; ram az i/o alá kapcsolva
,,003f9 sta $ff00    ; memória konfiguráció beállítása
,,003fc cli          ; megszakítás feloldva
,,003fd jmp $03ec    ; ugrás a program maradékára
,,003ec stx $ff00    ; régi konfiguráció beállítása
,,003ef rts          ; ugrás a főprogramra
```

A trükk abban áll, hogy a megszakítást meghatározott időre lezárjuk. A DMA-vezérlő normál esetben abban a pillanatban a RAM-bővítésben dolgozik, amikor a $ DF01 rekesz átíródik. Ha azonban a megszakítás le van tiltva, akkor a vezérlő csak a feloldó CLI után indul. Egészen eddig az utasításig bármely tetszőleges memória konfigurációt állíthatunk be. Ennek megfelelően a FETCH, STASH és SWAP utasítások kifogástalanul működnek.

A C64-es reset kivédése

A C64-es számos programja használja a $8004-es címen alkalmazható úgynevezett moduljelzést a resetállóság érdekében. A C128-as tulajdonosok ezt az akadályt könnyen átugorhatják. Váltsunk ki egy resetet a ⟨RUN/STOP⟩ gomb egyidejű megnyomása mellett. A C128-as a beépített monitorral jelentkezik vissza. Adjuk be az alábbi sort: ⟩08004 00

Ezzel a 64-es moduljelzést tönkretesszük. A 64-es módot kívánságra a J FFF4d utasítással érhetjük el, illetve egy resettel, ha közben nyomjuk a ⟨C=⟩ (Commodore) billentyűt. A 64-es mód a moduljelzést nem találván a beviteli móddal jelentkezik be. Adott esetben a kívánt programon máris dolgozhatunk.

PÖTYÖGŐ SZOLGÁLAT

296	+4	91/ 7	Logi Kód	60.-
297	C64	92/ 1	Databevivö	50.-
298	+4	91/ 7	Bomber	50.-
299	+4	91/ 7	Dominó - K	50.-
300	+4	91/ 7	The Wall	40.-
301	C64	91/ 7	Mozaik - K	50.-
302	+4	91/ 7	Dominó - R	60.-
303	+4	91/ 7	*Borgok Kincse	190.-
304	+4	91/ 7	Asteroids	60.-
305	C64	91/ 9	Ékezetes irat forrás	40.-
306	C64	91/ 9	Yugo2	50.-
307	C64	91/ 9	Adventure - Sz	130.-
308	C64	91/ 9	Morze Oktató B.	130.-
309	+4	91/ 9	Memory Game Plus	100.-
310	C64	91/ 9	*Capitaly	130.-
311	C64	91/ 9	*Gengszterek	60.-
312	C64	91/ 9	*Küldetés 2000	100.-
313	C64	91/ 9	Itt a $ hol a $	50.-
314	C64	91/ 9	Hewing	50.-
315	+4	91/10	Mikrobi	50.-
316	+4	91/10	Nest of Fleas	190.-
317	+4	91/10	Adventure - F	100.-
319	C64	91/10	Gondolkoz6	100.-
320	C64	91/10	Memori - B	130.-
321	+4	91/10	Matekastély	100.-
322	C64	91/11	Lottó	80.-
323	C64	91/11	Totó	80.-
324	C64	91/11	Német gyakorló	60.-
325	+4	91/11	Beszélö szótár	130.-
326	+4	91/11	File examiner	60.-
327	C64	91/11	Sprite szerkesztö	50.-
328	C64	91/11	Egyszinü Sprite	50.-
329	C64	91/12	Rulett javitás	50.-
330	C64	91/12	Tözsde	130.-
332	C64	92/ 1	Kalandlap	80.-
333	C64	92/ 1	Levéliró	80.-
334	C64	92/ 1	Képernyömaszkoló	130.-
335	C64	92/ 1	Sprite Kezelö	100.-
336	C64	92/ 1	SI - OPART	40.-
337	C64	92/ 2	Patience Quartet	130.-
338	C64	92/ 2	Monopoly	130.-
339	C64	92/ 2	Akasztófa	80.-
340	C64	92/ 2	Karaktertervezö	60.-
341	C64	92/ 3	String Array Manager	100.-
342	C64	92/ 3	Sprite Basic	50.-
343	C64	92/ 3	Screen Help	50.-
344	C64	92/ 3	Gyors validate	60.-
345	C64	92/ 3	BASIC SYNTAXER I.	190.-
346	C64	92/ 4	Tologató C64-re	100.-
347	C64	92/ 4	ABC rendezö	100.-
348	C64	92/ 5	*Matek + Római számok	100.-
349	+4	92/ 6	Tesztkészitö + értékelö	100.-
350	C64	92/ 3	BASIC SYNTAXER II.	190.-
351	+4	92/ 7	Három játék	110.-
352	+4	92/ 7	Szökés	80.-
353	C64	92/ 7	Toto	150.-
354	C64	92/ 7	DIGIT analizis program	110.-
355	+4	92/ 7	SCREEN DUMP	150.-
356	C64	92/ 9	UNISEQ filekezelö	130.-
357	C64	92/ 9	Graphic Copy	60.-
358	C64	92/10	Printfox Resizer	130.-
359	C+4	92/11	Tologató II.	150.-
360	C+4	92/12	Sprite Plus-4-en	130.-
361	C64	92/12	Változólista	150.-
362	C64,+4	93/ 2	Szomszéd játék	200.-
363	C64	93/ 2	Táblás játék	200.-
364	C64	93/ 2	SEDA	150.-
365	C64	93/ 2	415 * 200	120.-
366	+4	93/ 2	OTP hitel	100.-
367	C64	93/ 3	Catalog-Check(CTC)	150.-
368	C64	93/ 3	Pattogó golyók	200.-
369	C64	93/ 3	Beszoritós (DO-GUTI)	150.-
370	C64	93/ 3	*Zsirozás	200.-
371	C64	93/ 4	Attrib 64	50.-
372	C64	93/ 4	Táblázatnyomtató	80.-
373	C+4	93/ 4	Boritékkészitö	80.-
374	C+4	93/ 4	TV-óra	50.-
375	C64	93/ 4	Mini Domino	150.-
376	C64	93/ 4	Megathello	120.-
377	C64	93/ 4	Reversi	120.-
378	C64	93/ 4	Tengeri csata	120.-
379	C64	93/ 5	Videokazetták	80.-
380	C+4	93/ 5	DIN-karakterek	80.-
381	C64	93/ 5	DIRYDIRY	80.-
382	C64	93/ 5	Birodalom	150.-
383	C64	93/ 6	Csöamöba	100.-
384	C+4	93/ 6	Függvényábrázolás	150.-
385	C64	93/ 6	Amöba - E	100.-
386	C64	93/ 6	Bödy Building	60.-
387	C64	93/ 7	Colour BASIC	80.-
388	C64	93/ 7	Directory kezelö	60.-
389	C64	93/ 7	ötmezös KONO	80.-
390	C64	93/ 7	Fogócska	80.-
391	C+4	93/ 7	Klick - Klack	120.-
392	C64	93/ 9	Orarend	60.-
393	C64	93/ 9	Malomjáték	80.-
394	C+4	93/10	Magyar nyomtatás	120.-
395	C64	93/10	Pettyes szfinx	60.-
396	C64	93/11	Ku Damm óra	60.-
397	C64	93/11	Elásott kincs	80.-
398	C64	93/11	Turbo Sort	80.-

A *-gal jelölt programok csak lemezen rendelhetök !

Kérem, hogy a **Commodore Újság** legközelebbi számában jelenjen meg a következő szövegű apróhirdetés:

(minden kockába egy betűt írjon)

Tagsági szám: ....

Feladandó az újság címére:
Commodore Újság
Budapest, 1388. Pf. 86.

Alulírott megrendelem a következő programokat a PÖTYÖGŐSZOLGÁLAT-tól:

PROGRAM SORSZÁMA	ÁRA
1.	,– Ft
2.	,– Ft
3.	,– Ft
4.	,– Ft
5.	,– Ft
6.	,– Ft
7.	,– Ft
8.	,– Ft
9.	,– Ft
Összesen: db	,– Ft

A programokat a SZOLGÁLAT által biztosított
lemezre (99 Ft/db)
kazettára (40 Ft/db)
az általam küldött adathordozóra kérem.
(Kérjük a megfelelő szöveg aláhúzását!)
Postaköltség + munkadíj
A fizetés módja: személyesen — csekken — utánvéttel
(Kérjük a megfelelő szöveg aláhúzását!)
ÖSSZESEN:

A megrendelő neve: ....

Címe: ....

Az Országos Commodore Egyesület szolgáltatásai

Egyesületi tagoknak 20% kedvezmény:

VC—20 memóriabővítés 3—27 kByte-os:	kiépítéstől függő
C—16, C—116 memóriájának bővítése 64 kByte-ra:	3500 Ft
C—16 belső 16 kByte-os EPROM bővítés:	1450 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	2800 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	2000 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	3200 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4	5000 Ft
FÉK C—16, C—116, +4 potméteres sebességváltoztatás 0%-tól 100%-ig fokozatmentesen	2000 Ft
TTL IC-teszter (Cartridge+lemezen a program)	4300 Ft
+4, C—16, C—116 UNI—ROM modul különféle kiépítésekben:	
— 8 kByte SOFT—ROM	3400 Ft
— 16 kByte SOFT—ROM	4000 Ft
— 8 kByte SOFT—ROM 16 kByte EPROM	4400 Ft
— 16 kByte SOFT—ROM 16 kByte EPROM	5000 Ft
— 16 kByte EPROM	2200 Ft

Egyesület tagoknak 30% kedvezmény:

Speeddos (átkapcsolható) operációs rendszer beépítése (C64 átalakítás, lemezegység átalakítás + párhuzamos kábel)	5000 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése	900 Ft
PAGEFOX magyar ékezetes kiadvány-szerkesztő cartridge (a teljes A/4-es oldal kinyomtatásához 640 pont/soros nyomtató szükséges minimum, pl. Citizen 120D)	5500 Ft
FASTLOAD cartridge (lemezes gyorstöltő, másoló, monitor)	1500 Ft
TTL IC-teszter cartridge + program	4300 FT
288/256 Kbyte-os eprombank (vezérlő epprommal)	5000 Ft
Epromégető (2716-tól 27512-ig)	5000 Ft
C64-hez tároló oszcilloszkóp	8000 Ft
C64-bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
256 K RAM-diszk (256 Kbyte RAM-mal)	14000 Ft
256 K RAM-diszk (64 Kbyte RAM-mal)	9000 Ft
2×64 Kbyte-os cartridge igény szerinti programokkal feltöltve	4.300 Ft
64 Kbyte-os cartridge igény szerinti programokkal feltöltve	3000 Ft
Képújság (teletext) dekóder C-64-re	10000 Ft
1764-es RAM bővítő GEOS-hoz is használható	11250 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az epprommal együtt egységesen	700 Ft

Árainkat az alkatrészárak változásai befolyásolhatják.

A fenti bővítések megrendelhetők levélben az O.C.E. címén, valamint személyesen a havonta rendezendő klubdélelőttön, ahol rendszeres bemutatót is tartunk.

Postázás esetén 100 Ft postaköltség kerül felszámításra.

A kedvezmény igénybevételéhez az O.C.E. tagsági igazolvány bemutatása szükséges.



SyQuest TECHNOLOGY

Hivatalos magyarországi disztribútora:

NOVOTRADE SZERVIZ Kft.

Cím: 1053 Budapest, Henszlmann I. u. 9
Telefon: 117-4144 Telefax: 117-9692

Cserélhető lemezes winchester!

Szervizeinkben és szerződött viszonteladóinkon keresztül az alábbiakat forgalmazzuk:

SQ 555	44 MB-os $5\frac{1}{4}$	SCSI drive,	SQ 400 lemez	(44 MB)
SQ 5110	88 MB-os $5\frac{1}{4}$	SCSI drive,	SQ 800 lemez	(88 MB)
SQ 3105A	105 MB-os $3\frac{1}{2}$	AT-BUS drive,	SQ 310 lemez	(105 MB)
SQ 2542A	42 MB-os $2\frac{1}{2}$	AT-BUS drive,	SQ 240 lemez	(42 MB)

Számítógépszerviz !

*Személyi számítógépek és perifériák javítását, karbantartását továbbra is tőlünk rendelje meg! Gyors, pontos és megbízható! Örök garanciát kap, ha **átalánydíjas javítási és karbantartási** szerződésformánkkal védi nagyértékű gépei műszaki állapotát.*

Várjuk jelentkezését az alábbi szervizeinkben:

1083	Budapest,	Szigony u. 8.	Tel.: 1343-153
3525	Miskolc,	Fazekas u. 1–3.	Tel.: 46/321-488
5600	Békéscsaba,	Bartók B. u. 37.	Tel.: 66/327-195
6724	Szeged,	Csongrádi sgt. 76.	Tel.: 62/493-185
9700	Szombathely,	Szalonak u. 31.	Tel.: 94/314-519

NOVOTRADE SZERVIZ KFT.

SZERENCSEJÁTÉK RT

Ára: 156 Ft

ACOMP

SZÁMÍTÁSTECHNIKAI KFT.

1125 Budapest, Királyhágó utca 2.	*Tel.: 156-6790*	*Fax: 251-2385*
1135 Budapest, Szent László út. 74./A.	*Tel.: 149-6165*	*Fax: 251-2385*
1191 Budapest, Katica utca 9.	*Tel.: 147-0625*	*Fax: 177-9419*

Commodore Amiga 500	29 990 Ft
Commodore Amiga 500 Plus	29 990 Ft
Commodore Amiga 600	29 990 Ft
Commodore Amiga 1200	49 990 Ft
Commodore Amiga CD-32	49 990 Ft
A 600 / A 1200-hoz 40 MB HD kábellel	21 990 Ft
60 MB HD kábellel	24 990 Ft
80 MB HD kábellel	28 990 Ft
120 MB HD kábellel	36 990 Ft
Commodore Amiga 4000/030/4MB/120MB	209 000 Ft
Commodore 1940 Multisync Stereo Monitor	49 990 Ft
Commodore 1084s Stereo-Color monitor	29 990 Ft
Commodore 1085s Stereo-Color monitor	29 990 Ft
Commodore A-520 TV-Modulator	4 990 Ft
Commodore C-64 II VIDEOGAME SET	13 990 Ft
Commodore C-64 TERMINAL SET	19 990 Ft
Commodore 1541 II Floppy drive	9 990 Ft
Commodore MPS 1230 printer	23 990 Ft
512 Kb órás memóriabővítő	4 990 Ft
2.0 Mb órás memóriabővítő	9 900 Ft
1.0 Mb-os chip bővítő Amiga 500 Plus-ba	6 500 Ft
1.0 Mb-os órás chip bővítő Amiga 600-ba	6 900 Ft
Mouse pad	220 Ft
Noris üveg 14" monitorfilter	1 490 Ft
Noris hálós 14" monitorfilter	400 Ft
Noris MB 80 3,5" lemeztartó	420 Ft
Noris DB 100 5,25" lemeztartó	420 Ft
Noris Porvédő Amiga 500 / 500 Plus	890 Ft
Noris Porvédő C-64 II.	590 Ft
Sound Blaster PRO-2 Deluxe	17 900 Ft
Sound Blaster 16 ASP Deluxe	32 900 Ft
Sound Blaster 16 Basic	24 900 Ft
Amiga Magazin (német) újság	490 Ft
Power Play (német) újság	490 Ft
3M DC2120 Streamer kazetta	2 900 Ft
Amiga – EuroScart kábel	1 190 Ft
NoName 3.5" DSDD lemez	550 Ft
NoName 3.5" DSHD lemez	790 Ft
NoName 5.25" DSDD lemez	200 Ft
Maxell 3.5" MF2-DD lemez	750 Ft
Maxell 3.5" MF2-HD lemez	1 390 Ft
Maxell 5.25" MD2-HD lemez	790 Ft
Verbatim 3,5" DSDD lemez	750 Ft
Verbatim 5,25" DSHD lemez	750 Ft
Profex 3.5" DSHD lemez (11 db/Form)	1 190 Ft
Profex 5.25" DSHD lemez (11 db/Form)	395 Ft
Fuji 5.25" MD 2DD lemez	330 Ft
BASF 5,25" MF 2DD lemez	330 Ft

Action Replay MK III. (Amiga) + könyv	9 990 Ft
Action Replay MK VI. Pro (C-64) + könyv	5 900 Ft
C-64 midi szoftverrel	6 500 Ft
C-64/C-128 Joystick Mouse	1 990 Ft
C-64/C-128 C1351 Mouse	2 490 Ft
Swifty Amiga/Atari Mouse (3 gombos)	1 990 Ft
4 Player Adapter(4 Joystick csatoló)	1 890 Ft
Képdigitalizáló + RGB splitter	13 900 Ft
Mouse — Joystick autómatikus kiválasztó	1 990 Ft
Real Time Clok for Amiga 1200	2 490 Ft
Midi Amiga Interface	3 490 Ft
Handyscanner Amigához	14 900 Ft
Boot Selector Amigához	1 490 Ft
Stereo hangdigitalizáló Amigához	6 490 Ft
Trackball Amigához	3 590 Ft
Rocgen Genlock for Amiga	9 900 Ft
Rockey Advanced Video Keying for Amiga	24 900 Ft
1.76 MB HD külső drive Amigához (OS2+)	16 990 Ft
Roctec 3,5" Amiga slim külső dirve	9 990 Ft
Roctec 3,5" Ivory Anti-Vírus Amiga drive	9 990 Ft
Roctec 3,5" Black Anti-Vírus Amiga drive	9 990 Ft
Roctec 3,5" Amiga belső drive	8 500 Ft
Rochard HD kontroller A500/A500+	17 900 Ft
+ 80 MB Hard Disk	24 990 Ft
+ 1 MB SIMM Ram	6 500 Ft
Joystickok	
Quickshot QS – 113 IBM	990 Ft
Quickshot QS – 123 Warrior IBM	1 290 Ft
Quickshot QS – 146 Intruder5 IBM	2 790 Ft
Quickshot QS – 151 Aviator5 IBM	3 490 Ft
Quickshot QS – 101 I.	450 Ft
Quickshot QS – 102N II.	580 Ft
Quickshot QS – 102P II. Plus Mikrokap.	660 Ft
Quickshot QS – 111A II. Turbo Mikrokap.	790 Ft
Quickshot QS – 128 Maverickl	1 390 Ft
Quickshot QS – 129F Flightgripl	750 Ft
Quickshot QS – 130F Python1	850 Ft
Quickshot QS – 137F PythonIM Mikrok.	990 Ft
Quickshot QS – 155 Aviatorl Mikrokap.	2 890 Ft
Quickshot QS – 149 Intruderl	2 690 Ft
Quickshot QS – 130N Pyton2 Nintendo	990 Ft
Dynamics Competition Pro 5000	1 090 Ft
Dynamics Competition Pro IBM	3 990 Ft
Dynamics Competition Pro 5000 Mini	1 490 Ft
Dynamics Competition Star Mini	1 990 Ft
Dynamics Competition Special Mini	1 690 Ft
Dynamics Competition Transparent Mini	1 690 Ft

Áraink mindenkori változtatásának a jogát fenntartjuk!

Áraink az 1 év garanciát és az ÁFA-t tartalmazzák!

SZENZÁCIÓ!! SEGA MEGADRIVE + 1 CONTROLE PAD + 1 JÁTÉK KÁRTYA **CSAK 18 990 Ft**

SZERENCSEJÁTÉK RT